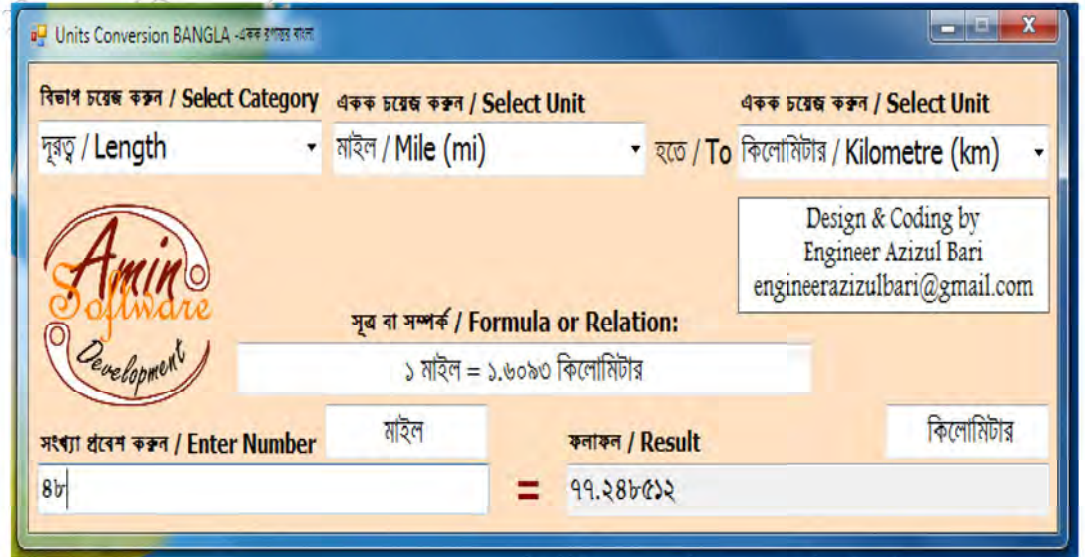




ভিজুয়াল বেসিক.নেট প্রোগ্রামিং

নতুন শিক্ষার্থীদের জন্য গুরুত্বপূর্ণ প্রশিক্ষণ রিসোর্স

প্রথম খণ্ড Part ONE



বিশেষ ইন্টারনেট সংস্করণ Special Internet Edition

Copyright NOTICE:

All rights reserved. This electronic book or any portion thereof may not be reproduced or used in any manner whatsoever without the express written permission of the author except for the use of brief quotations in a book review. This electronic book may only be used for private reading, it is forbidden to print and distribute all or any part of it and distribute for commercial purposes.

ইঞ্জিনিয়ার আজিজুল বারী
engineerazizulbari@gmail.com

খানক্বায়ে আমীনিয়া-আসগরিয়া গবেষণা বিভাগ
Ali Centre, Subidbazar, Sylhet, BD



বিষয়সূচি



১ম পরিচ্ছেদ: ভূমিকা: মাইক্রোসফট ভিজুয়াল বেসিক.নেট - কী এটা?	৪
২য় পরিচ্ছেদ: শুরু করুন প্রাকটিস: কী লাগবে প্রাকটিস করতে?	৫
১ম প্রোগ্রাম: ProjectBismillah	৬
User Interface ডিজাইন	৮
FormBorderStyle ও MaximumBox ডিজাইনে পরিবর্তন	৯
Designer থেকে Programmer	১০
৩য় পরিচ্ছেদ: ভিজুয়াল বেসিক.নেট ভাষার মৌলিক কাঠামো	১২
ভিজুয়াল বেসিক.নেট : Subs & Functions	১২
১ম উদাহরণ: Sub	১৩
২য় উদাহরণ: Function	১৩
৩য় উদাহরণ: Sub & Function	১৬
৪র্থ পরিচ্ছেদ: ভিজুয়াল বেসিক.নেট ভাষার Variable Declaration	২১
ভিজুয়াল বেসিক.নেট : Variable & Data Type	২১
CollectionDemo Application	২৭
৫ম পরিচ্ছেদ: Looping in Program	৩০
বিভিন্ন লুপিং : Various Looping	৩০
৬ষ্ঠ পরিচ্ছেদ: Structure	৩২
বিভিন্ন ভেরিয়েবল একত্রে : Different Variables together	৩২
৭ম পরিচ্ছেদ: Separate code block in Program	৩৬
মডিউল ও ক্লাস : Module & Class	৩৬
৮ম পরিচ্ছেদ: More on Subs & Functions	৪০
সাব ও ফাংশন : Subs & Functions	৪০
NumbersAdder উদাহরণ	৪৪
৯ম পরিচ্ছেদ: Ready-made functions and methods	৪৬
বানানো ফাংশন ও মেথড: Ready-made functions & Methods	৪৬
১০ম পরিচ্ছেদ: Object Oriented Programming [OOP] -বস্তুভিত্তিক প্রোগ্রামিং	৫৫
বস্তুভিত্তিক প্রোগ্রামিং: Object Oriented Programming [OOP]	৫৫
ক. Encapsulation	৫৫
খ. Inheritance	৫৬
গ. Polymorphism	৫৭
১১তম পরিচ্ছেদ: Working with Files	৬১
ফাইল নিয়ে কাজ : Working with Files	৬১
উদাহরণ-১: My.Computer.FileSystem Object ব্যবহার করে ফাইল নিয়ন্ত্রণ	৬৫
১২তম পরিচ্ছেদ: Accessing Computer Resources	৭১
কম্পিউটারের রিসোর্স সংগ্রহ ও ব্যবহার : Accessing Computer Resources	৭১
My.Computer.Object	৭১
উদাহরণ-১: Simple .wav audio player -.wav ফাইল অডিও প্লেয়ার	৭১





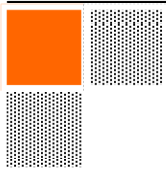
বিষয়সূচি



১৩তম পরিচ্ছেদ: Software Development Cycle	৮০
সফটওয়্যার তৈরির প্রক্রিয়াচক্র : Software Development Cycle	৮০
ক. প্রয়োজন সনাক্তকরণ - Identify Requirement	৮০
খ. বিশ্লেষণ - Analysis , গ. আকল্প - Design	৮০
ঘ. সংক্ষেতলিপি - Coding , ঙ. পরীক্ষণ - Testing , চ. বিস্তারণ - Deployment	৮১
১৪তম পরিচ্ছেদ: Complete Application Tutorials	৮২
পূর্ণাঙ্গ এ্যাপলিকেশন প্রশিক্ষণ : Complete Application Tutorials	৮২
Application 1: BD City to City Distance - বাংলাদেশ শহর থেকে শহর দূরত্ব	৮২
Application 2 : Bangla Calculator - বাংলা ক্যালকুলেটর	৯১
Application 3 : Bangla / English Unit Conversion Software - বাংলা / ইংরেজি একক পরিবর্তন সফটওয়্যার।	৯৫
Application 4 : Bangla / English Word Processor - বাংলা / ইংরেজি ওয়ার্ড প্রসেসর।	১২৭
১৫তম পরিচ্ছেদ: Database Programming Made Simple	১৩৯
সহজ ডাটাবেইজ প্রোগ্রামিং: Database Programming Made Simple	১৩৯
১. এন্টিভএক্স ডাটা অবজেক্ট.নেট [সংক্ষেপে: ADO.Net]	১৩৯
১. ভিজুয়াল বেসিক.নেট ও ডাটাবেইজ: একটি ঠিকানা বই ডাটাবেইজ প্রোগ্রাম	১৪১
ডাটাবেইজ প্রোগ্রামিংয়ের ভাষা: Structured Query Language [SQL]	১৪৬
১৬তম পরিচ্ছেদ: Few More Sample Applications	১৬৫
আরো কয়েকটি নমুনা এ্যাপলিকেশন: Few More Sample Applications	১৬৫
১. একটি ব্যক্তিগত ওয়েব ব্রাউজার: A Personal Web Browser	১৬৫
২. সাধারণ এ্যানিমেশন: Simple Animation	১৭০
৩. বাংলা-ইংরেজি-বাংলা তারিখ রূপান্তর: Bangla-English-Bangla Date Conversion	১৭১
১৭তম পরিচ্ছেদ: Windows Form Controls	১৮৪
উইন্ডোজ ফর্ম কন্ট্রোলস: Windows Form Controls	১৮৪
১৮তম পরিচ্ছেদ: Visual Basic.Net: Simple Guide	২১১



১ম পরিচ্ছেদ: ভূমিকা



মাইক্রোসফট ভিজুয়াল বেসিক.নেট - কী এটা?

এটা একটি সহজ কম্পিউটার প্রোগ্রামিং ভাষা। যে কেউ ইচ্ছে, এ ভাষা শিখে বিশ্বয়কর সফটওয়্যার তৈরি করতে পারে পি.সি কম্পিউটারের জন্য। এ ভাষার মৌলিক ক'টি বৈশিষ্ট্য:

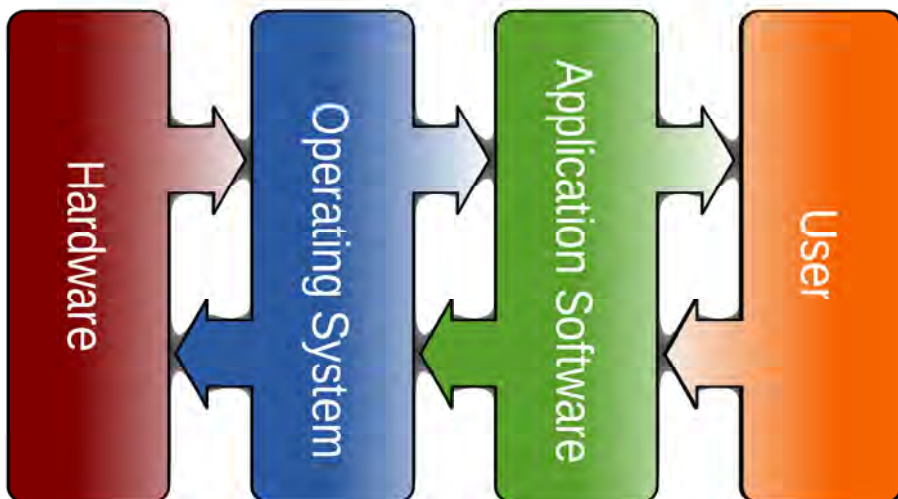
১. এটা বস্তুভিত্তিক প্রোগ্রামিং ভাষা (Object Oriented Programming Language)।
২. এটা আধুনিক, সহজ, জনপ্রিয়, ইংরেজির মতো ভাষা।
৩. এটা কাঠামোসর্বস্ব ভাষা (Structured Language)।
৪. এ ভাষায় লেখা প্রোগ্রামগুলো খুব নির্ভরশীল।
৫. ভুল-ত্রুটি সঠিককরণ (Debugging) অনেকটা সহজ।
৬. এটা মাইক্রোসফটের ডট.নেট ফ্রেমওয়ার্ক (.Net Framework) এর অংশ।
৭. এ ভাষায় লিখিত সফটওয়্যার মাইক্রোসফট উইন্ডোজ এর জন্য খুব মানানসই।

জানেন কি?: পৃথিবীবিশ্বব্যাপী মাইক্রোসফট অফিস এর প্রতিটি এ্যাপলিকেশন ভিজুয়াল বেসিক (এবং ভিজুয়াল বেসিক.নেট) দ্বারা নিয়ন্ত্রণ ও রদবদল (customize) করা যায়।

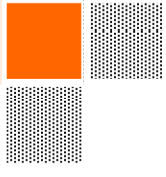
এ পর্যন্ত যা বলা হলো তার এক বা একাধিক বিষয় আপনার বুঝে না আসলে ভাববেন না। এই গ্রন্থ ভালো করে অধ্যয়ন ও এতে সম্পূর্ণ বিভিন্ন পদে পদে বর্ণিত মডেল প্রোগ্রামগুলোর ওপর প্রাকটিস করে নিলে আপনিও হয়ে যাবেন ভিজুয়াল বেসিক.নেট প্রোগ্রামার। আপনাকে প্রোগ্রামার বানানোই আমাদের ইচ্ছে। পাঠ করতে থাকুন। দেখুন, কম্পিউটার প্রোগ্রামিং এর বিশ্বয়কর জগৎ! এ বিশ্বয়ের জগতে পাড়ি দিতে আপনার হাতিয়ার শুধুমাত্র একটি ল্যাপটপ বা ডেস্কটপ কম্পিউটার ও সেসাথে শেখার স্পৃহা- এই যা!

এই গ্রন্থে আপনি যা শিখবেন:

১. ভিজুয়াল বেসিক.নেট ভাষার মাধ্যমে সফটওয়্যার তৈরির পদ্ধতি।
২. উইন্ডোজের .exe এ্যাপলিকেশন তৈরি।
৩. উইন্ডোজের বেসিক ডাটাবেইজ প্রোগ্রামিং।



২য় পরিচ্ছেদ: শুরু করুন প্রাকটিস



কী লাগবে প্রাকটিস করতে?

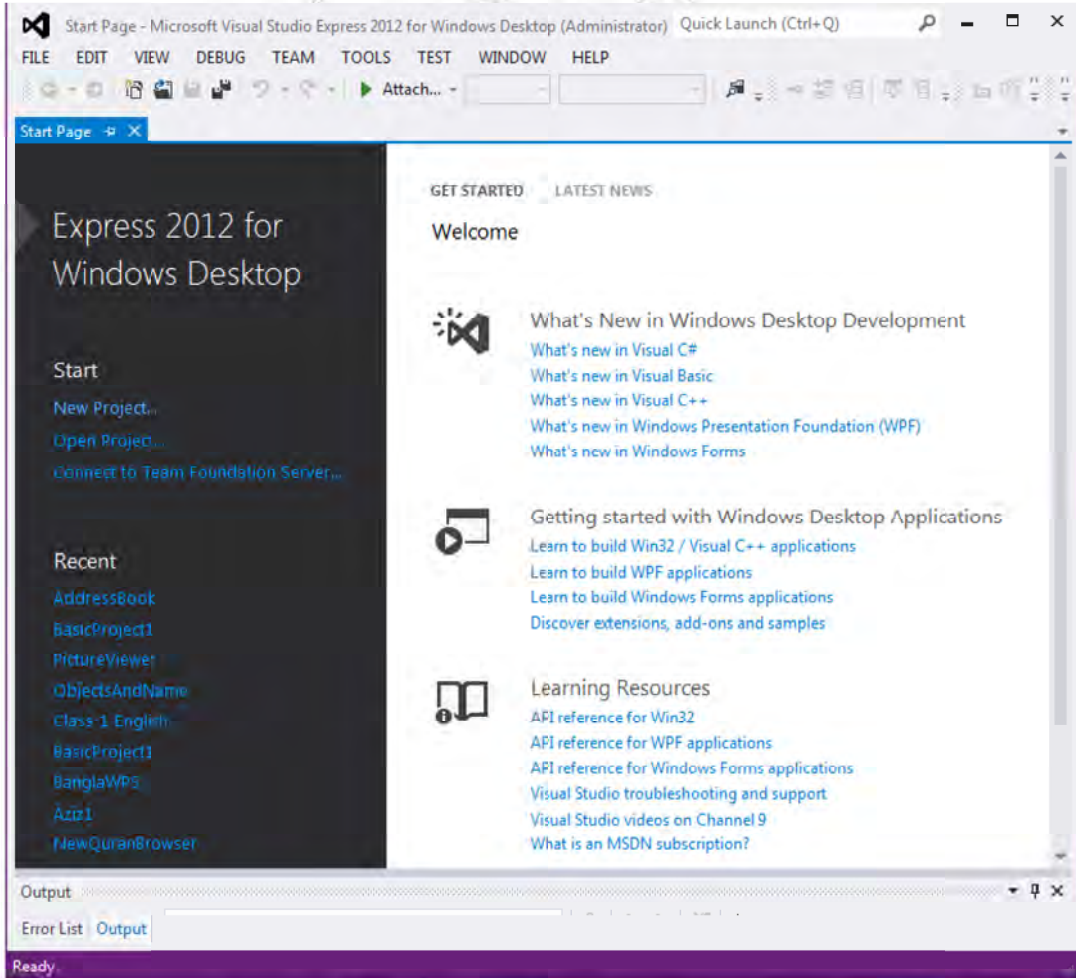
আই.ডি.ই (Integrated Development Environment)

বাংলায় বলা যায়, ‘অঙ্গীভূত উন্নয়ন পরিবেশ’। মূলত আই.ডি.ই হলো সফটওয়্যার তৈরির একটি সফটওয়্যার। এতে আছে বিভিন্ন ধরনের হাতিয়ার (Tools) যা ব্যবহারের মাধ্যমে আপনি সফটওয়্যার সৃষ্টি করতে সক্ষম হবেন। আপনার সুবিধার্থে মাইক্রোসফট একাধিক উন্নতমানের আই.ডি.ই তৈরি করেছে। আপনার পছন্দসই আই.ডি.ই ইন্টারনেট থেকে (অথবা কম্পিউটারের দোকান থেকে) কম্পিউটারে ইনস্টল করতে পারেন। এজন্য আপনাকে তেমন বেশি টাকা দিতে হবে না। ইন্টারনেট থেকে ফ্রি ডাউনলোড করতে পারেন।

এই গ্রন্থে আমি মাইক্রোসফটের ‘ভিজুয়াল স্টুডিও এক্সপ্রেস ২০১২ ফর ডেস্কটপ’ থেকে উদাহরণ দিয়েছি। আপনি এটা ডাউনলোড করতে পারেন এই ওয়েব ঠিকানা থেকে:

<https://www.microsoft.com/en-us/download/details.aspx?id=34673>

ইনস্টল করার পর মাইক্রোসফটের ওয়েবসাইট থেকে বিনামূল্যে রেজিস্ট্রেশন করুন। এরপর ভিজুয়াল বেসিক এক্সপ্রেস ফর ডেস্কটপ যখনই রান করবেন তখন প্রথমে এই উইন্ডোটি আসবে:



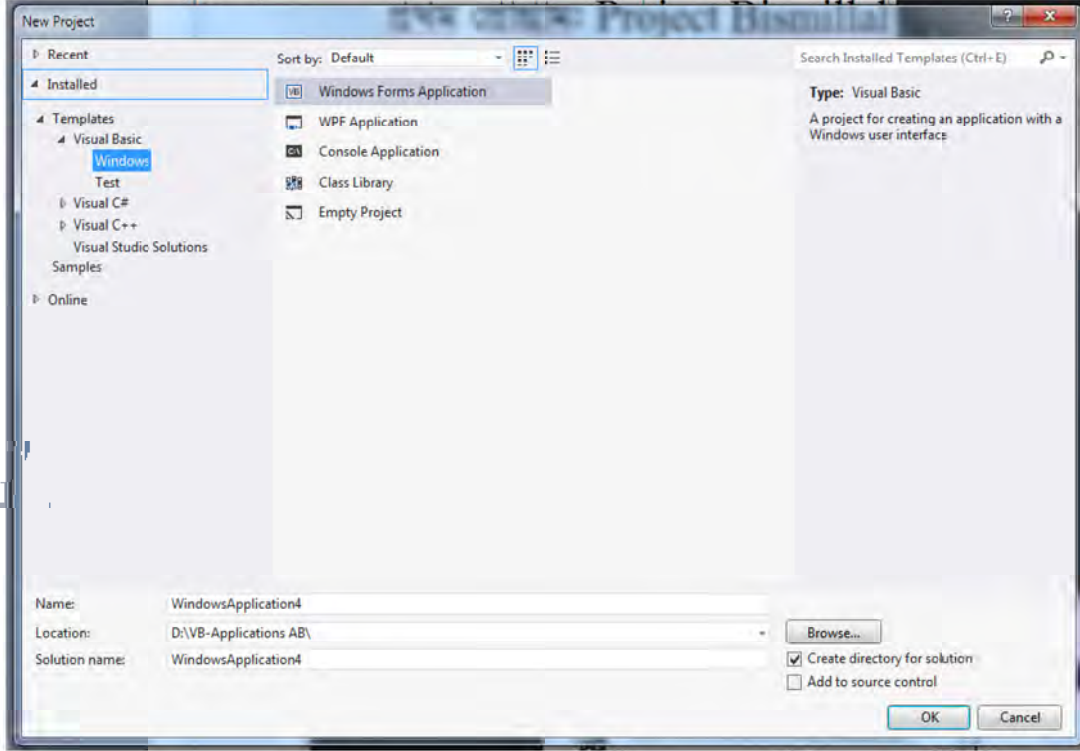
চিত্র-১: এক্সপ্রেস ২০১২ ফর উইন্ডোজ ডেস্কটপ রান করার পর প্রথমে প্রদর্শিত উইন্ডো।

১ম প্রোগ্রাম: ProjectBismillah

নিচে ধারাবাহিকভাবে বর্ণিত নির্দেশনাবলী অনুসরণ করে প্রজেক্ট তৈরি করুন:

১. Start এর নিচে লিখিত New Project... -এ ক্লিক করুন।

নতুন একটি প্রজেক্ট তৈরির জানালা আসবে। এটি নিম্নে দেখানো হলো।



চিত্র-২: New Project... -এ ক্লিক করার পর প্রদর্শিত জানালা।

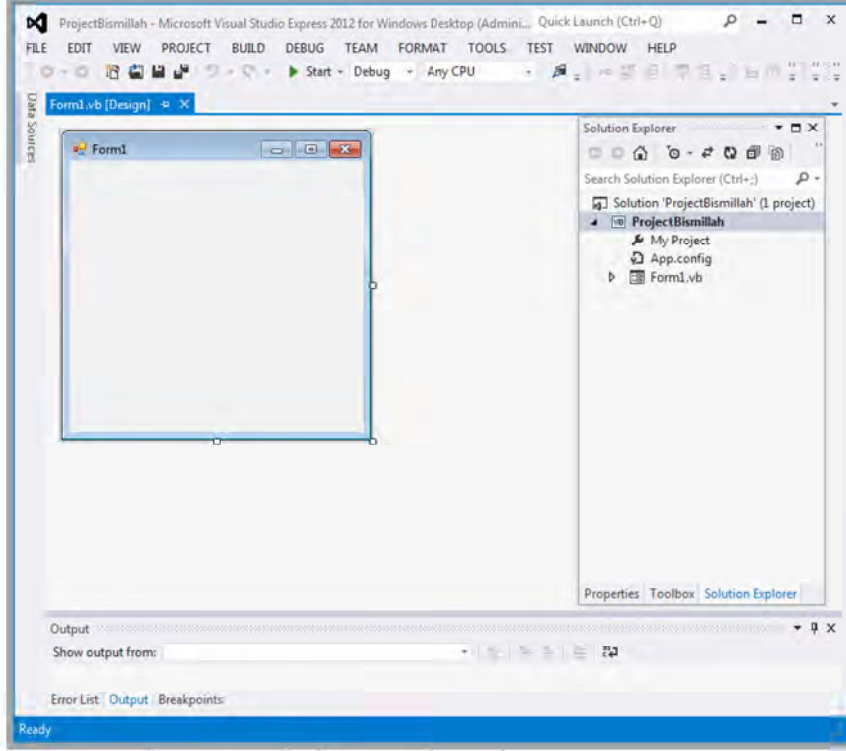
আপনি ডেস্কটপে LearnigDotNet নামকরণে একটি ফন্ডার বানিয়ে এর ভেতর সকল Project সংরক্ষণ করতে পারেন।

২. নতুন প্রজেক্ট ProjectBismillah -সংরক্ষণ করুন।

জানালায় Name ফিল্ডে লিখুন ProjectBismillah। Windows Forms Application যেনো হাইলাইট (ক্লিক করা) থাকে। আমরা এই গ্রুপে এরূপ প্রোগ্রামই করবো। Browse...-এ ক্লিক করে আপনার পছন্দসই ফন্ডারে সংরক্ষণের জন্য চয়েজ করুন। এরাব ক্লিক করুন 'Ok' বোতামে। আপনার নতুন প্রজেক্ট তৈরি হবে। কিছুক্ষণ পর প্রজেক্ট খুলবে এবং পরের পৃষ্ঠায় চিত্রিত জানালাটি প্রদর্শিত হবে।

ব্যস! আপনি প্রথম প্রজেক্ট তৈরি করে নিলেন। এবার আসুন, এই প্রজেক্টের উদ্দেশ্য হাসিলে কাজে লেগে যাই- কেমন?

১ম প্রোগ্রাম: ProjectBismillah



চিত্র-৩: ProjectBismillah তৈরির পর প্রদর্শিত আই.ডি.ই জানালা।

উপরের চিত্রের মতো আপনার কম্পিউটারে প্রদর্শিত আই.ডি.ই ঠিক হুবহু একই না-ও হতে পারে। কোনো ভাবনা নেই। যেভাবেই আসুক না কেনো, এতে সর্বদাই Form1 নামকরণে একটি ফর্ম থাকবে। এই ফর্ম বা জানালাটির ভেতরই আমাদের প্রোগ্রামের যাবতীয় বস্তু তৈরি হবে। একটু পরই আপনি বিষয়টি বুঝতে সক্ষম হবেন।

৩. সলিউশন এক্সপ্লোরার প্রদর্শন করুন।

যদি Solution Explorer জানালা প্রদর্শিত না হয়ে থাকে তাহলে VIEW ম্যানুতে ক্লিক করে চতুর্থ আইটেম Solution Explorer চয়েজ করুন। এটি টেনে ডানদিকে নিয়ে আসুন।

৪. টুলবক্স প্রদর্শন করুন।

যদি Toolbox জানালা প্রদর্শিত না হয়ে থাকে তাহলে VIEW ম্যানুতে ক্লিক করে ত্রয়োদশ আইটেম Toolbox চয়েজ করুন। এটি টেনে ডানদিকে নিয়ে আসুন। চাইলে Solution Explorer জানালার ভেতর ছেড়ে দিতে পারেন।

৫. প্রোপার্টিজ প্রদর্শন করুন।

যদি Properties জানালা প্রদর্শিত না হয়ে থাকে তাহলে VIEW ম্যানুতে ক্লিক করে শেষেরটির আগের আইটেম Properties Window চয়েজ করুন। এটি টেনে ডানদিকে নিয়ে আসুন। চাইলে Solution Explorer জানালার ভেতর এটাও ছেড়ে দিতে পারেন।

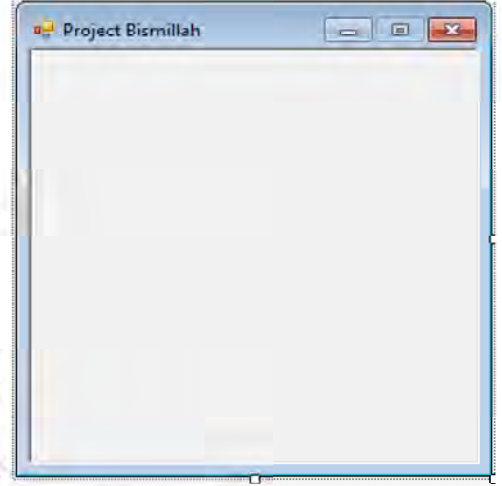
উপরে বর্ণিত উপায়ে কাজ করলে যে জানালাটি দেখা যাবে তা হবে এ পৃষ্ঠার চিত্রের মতো (চিত্র-৩)। লক্ষ করুন Solution Explorer জানালার নিচে তিনটি বোতাম আছে: Properties, Toolbox এবং Solution Explorer। যে সময় যেটির প্রয়োজন তাতে ক্লিক করলেই সে জানালা প্রদর্শিত হবে। আপাতত আমাদের দরকার এই তিনটি ও Form1 নামক জানালাটি।

ব্যস! ProjectBismillah -এর জন্য আপনি আই.ডি.ই প্রস্তুত করে নিয়েছেন। এবার কিছু প্রোগ্রামিং হবে।

FormBorderStyle ও MaximumBox ডিজাইনে পরিবর্তন

প্রোগ্রাম চলার সময় যাতে ফর্ম জানালার আয়তন পরিবর্তন না হয় সে জন্য দু'টি প্রোপার্টিতে পরিবর্তন করলে ভালো হবে। এই দু'টি পরিবর্তন করতে নিম্নের ধারাবাহিক পদক্ষেপগুলো অনুসরণ করুন:
ফর্মে ক্লিক করে এটাকে সিলেক্ট করুন। ক্লিক করুন সলিউশন এক্সপ্লোরার জানালায়। এবার স্ক্রল করে FormBorderStyle -এ যান। ডানের ড্রপডাউন মেনু থেকে চয়েজ করুন Fixed3D আইটেম। এরপর স্ক্রল করে চলে যান MaximumBox -এ। এর ডানের বাক্সে True লেখা থাকলে ড্রপডাউন বাক্স থেকে False শব্দটি চয়েজ করুন।

এবার আপনার ফর্মের চেহারা ডানের চিত্রের মতো হবে। প্রোগ্রাম Start করে দেখুন- ফর্মের সাইজ পরিবর্তন করা আর সম্ভব হবে না।



চিত্র-৬: Form1 -এর চিত্র।

একটি Button ও একটি Text box ফর্মে যুক্তকরণ

আমরা ফর্মের ডিজাইন শেষ করেছি। এবার এর ভেতর দু'টি বস্তু যুক্ত করবো। প্রথমে একটি Text box ফর্মের উপরে এনে বসাবো। নিচে বর্ণিত পদক্ষেপগুলো ধারাবাহিকভাবে অনুসরণ করুন।
ক্লিক করুন সলিউশন এক্সপ্লোরার জানালায়। ক্লিক করুন নিচের Toolbox বাক্সে। Common Controls -এর নিচে দেখতে পাবেন অনেকগুলো আইটেম। ডবল ক্লিক করুন ২নং আইটেম Button-এ। দেখবেন, Button1 -নামক একটি বোতাম ফর্মের উপরের বায়ের কোণে চলে গেছে। আপনি বোতাম যোগ করে নিয়েছেন।

এবার টুল বাক্সে যেয়ে স্ক্রল করে নিচে যান। আগের মতো ডবল ক্লিক করুন TextBox -নামক আইটেমে। লক্ষ করুন, একটি বাক্স চলে গেছে ফর্মের ভেতর। এটাকে মাউস দিয়ে টেনে মাঝখানে নিয়ে আসুন। বোতামটিও নিচে সেন্টারে নিয়ে আসুন।

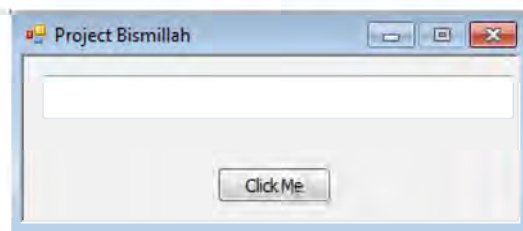
টেক্সট বাক্স ও বোতামের প্রোপার্টি পরিবর্তন

আমরা চাই টেক্সট বাক্সের লেখাটুকু একটু বড় সাইজ হবে। এছাড়া বোতামে লেখা থাকবে Click Me শব্দগুলো। লেখার সাইজও কিছুটা বড় হবে। আমরা এ কাজগুলো সহজে করতে পারি প্রোপার্টিজ জানালা ব্যবহার করে। নিম্নে বর্ণিত পদক্ষেপগুলো অনুসরণ করুন।

Button - ক্লিক করে Button1 সিলেক্ট করুন। এক্সপ্লোরার জানালার নিচে Properties বাক্সে ক্লিক করুন। এবার প্রথম দিকের + Font -এর + চিহ্নে ক্লিক করুন। আরো একটি মেনু আসবে। এখানে Size কথাটির বাক্সে ৮ মুছে ১৪ লিখুন। প্রোপার্টিজ জানালার ডানের স্ক্রলে ধরে নিচের দিকে যেয়ে Text আইটেম বের করুন। এর ডানে Button1 শব্দটি মুছে Click Me শব্দদ্বয় লিখুন। ব্যস! বোতামের পরিবর্তন সম্পূর্ণ হলো।

TextBox - ক্লিক করে টেক্সট বাক্স সিলেক্ট করুন। প্রোপার্টিজ জানালায় যেয়ে আগের মতো ফন্ট সাইজ বদলে ১৪ করুন। ব্যস! হয়ে গেল।

ফর্মের সাইজ পরিবর্তন, টেক্সট বাক্স ও বোতামের অবস্থান পরিবর্তন শেষে যে ফর্ম তৈরি হলো তা নিম্নে দেখানো হয়েছে।



চিত্র-৭: Form1 -এর শেষ চিত্র।

১ম প্রোগ্রাম: ProjectBismillah

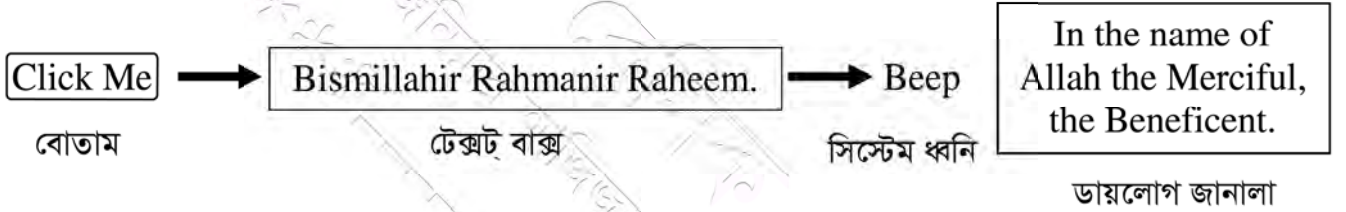
Designer থেকে Programmer

এ পর্যন্ত আপনি যা করলেন তাতে আপনাকে প্রোগ্রামার উপাধিতে ভূষিত করা যাবে না! হ্যাঁ, আপনি ডিজাইনার ছিলেন। কারণ আপনি একটি ইউজার ইন্টারফেইস ডিজাইন করেছেন সফলভাবে। তবে আমাদের ইচ্ছা আপনি শুধু ডিজাইনার নয় একজন দক্ষ প্রোগ্রামার হবেন। অর্থাৎ বুঝে বুঝে আপনি প্রোগ্রাম চলাকালের এ্যাকশন কোড লিখার যোগ্যতা লাভ করবেন। আর এটাই তো এ গ্রন্থ রচনার উদ্দেশ্য।

প্রোগ্রাম করা যেমন কঠিন মনে করা হয় বাস্তবে তা কিন্তু নয়। অবশ্য এজন্য মাইক্রোসফটকে ধন্যবাদ দিতেই হবে। এক সময় এমন ছিলো, আপনি যে ইউজার ইন্টারফেইস কয়েক মিনিট বা ধরণ ঘণ্টা খানেকের মধ্যেই তৈরি করে নিলেন তা করতে, লিখতে হতো শত শত লাইনসর্বস্ব কোড। সময় লাগতো কয়েক দিন পর্যন্ত। কিন্তু এ কাজগুলো আমরা রেডি-মেইড পেয়ে যাচ্ছি। যা তারা রেডি-মেইড করে দিতে পারেন নি তাহলো আমাদের চাহিদা। কারণ তারা তো জানেন না, প্রোগ্রাম দ্বারা আমরা কি করতে চাই? এ কারণেই বাকী অংশ নিজেরাই করতে হবে।

প্রোগ্রাম কোডিং

আমরা চাই, যে মুহূর্তে ব্যবহারকারী Click Me বোতামে ক্লিক করবেন, তখনই কিছু তথ্য প্রদর্শন করতে। কিসব তথ্য প্রদর্শিত হবে তা নিম্নে চিত্রিত হয়েছে।



উইন্ডোজ ফরম এপ্লিকেশনের কোডিং ফিল্ডে সর্বদাই নিম্নে চিত্রিত একটি কোড ব্লক থাকে। এই ব্লকের ভেতরেই যাবতীয় কোড লিখতে হবে।

Public Class Form1

সব কোড এখানে

End Class

Click Me - বোতামের কোড

ডিজাইন উইন্ডোতে চিত্রিত বোতামে ডবল ক্লিক করলেই নিচের ফিল্ডটি আসবে। আমাদের লেখা লাইনটি আন্ডারলাইন করে দেখানো হয়েছে। মূল কোডে আন্ডারলাইন থাকবে না।

Public Class Form1

```

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    TextBox1.Text = "Bismillahir Rahmanir Raheem."
End Sub
  
```

End Class

এই `TextBox1.Text = "Bismillahir Rahmanir Raheem."` লাইন কোড দ্বারা আমরা নির্দেশ দিচ্ছি টেক্সট বাক্সে যা লিখা হবে।

১ম প্রোগ্রাম: ProjectBismillah

আমরা এবার Click Me বোতামের Sub-এ আরো দু'টি লাইন যুক্ত করে নেবো। লাইন দুটো হলো:

```
Beep()  
MsgBox("In the name of Allah the Merciful, the Beneficent.")
```

ব্যস! কোডিং শেষ হলো। Form1.vb* -নামক ফিল্ডে এবার যা লিখা থাকবে তাহলো:

Public Class Form1

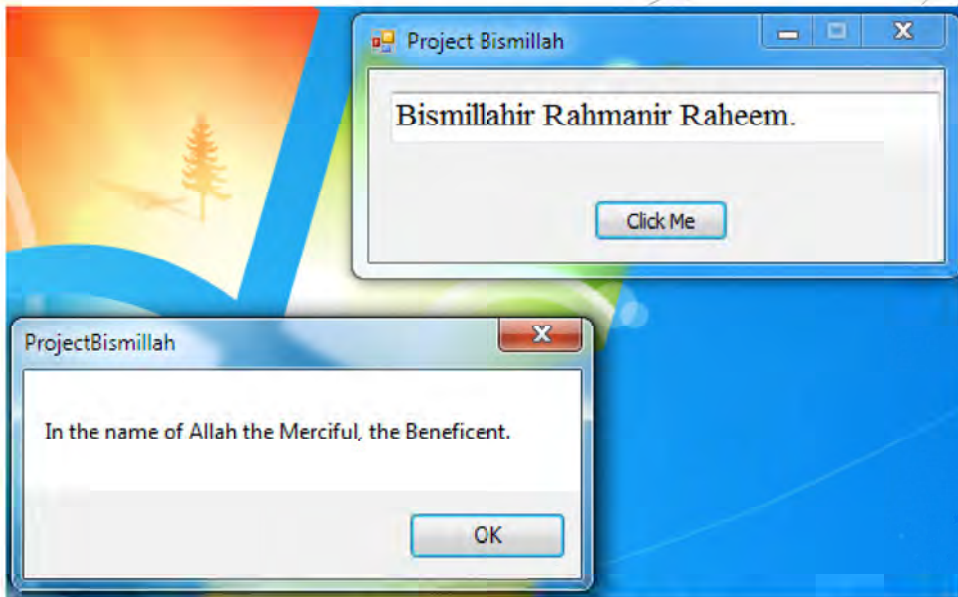
```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click  
    TextBox1.Text = "Bismillahir Rahmanir Raheem." 'what we want to be written in Text Box.  
    Beep() 'System sound.  
    MsgBox("In the name of Allah the Merciful, the Beneficent.") 'The text for message box window.  
End Sub  
End Class
```

ব্যাখ্যা: Click Me বোতামের Click নামক একটি Event আছে। আমরা এই ইভেন্টটি ধরে কম্পিউটারকে ৩টি কোড লাইনে নির্দেশ দিচ্ছি। যথা: লাইন-১ দ্বারা বলা হচ্ছে ফর্মের TextBox1-এ কী লেখা হবে। লাইন-২ দ্বারা নির্দেশ দিচ্ছি কম্পিউটারের Beep নামক ধ্বনিটি প্লে করার জন্য। এটি সিস্টেমে সংরক্ষিত আছে। লাইন-৩ দ্বারা বলা হচ্ছে একটি মেসেজ বাক্স খুলে তাতে কিছু লিখে প্রদর্শন করতে।

দ্র: লক্ষ করুন, কোডে সবুজ কিছু লেখা আছে। এগুলোর শুরুতে ' -এ সিঙ্গেল কোট মার্কও দেওয়া হয়েছে। এ সবুজ লাইনগুলো হলো 'কমেন্ট'। ভিজুয়াল বেসিক এগুলো সম্পূর্ণরূপে উপেক্ষা করে যাবে। প্রোগ্রামারের সুবিধার্থে এই কমেন্ট লাইনের ব্যবস্থা। মন্তব্য করলে কোডের অর্থ অনেকটা স্পষ্ট হয়ে ওঠে। আপনিও অবশ্যই মন্তব্য করবেন। এগুলো রেডি-হওয়া প্রোগ্রামের কোনো ক্ষতি করে না কিংবা এদের কারণে অতিরিক্ত মেমোরিও খরচ হয় না। তবে প্রয়োজনে প্রোগ্রাম সঠিককরণের [ডিবাগিং] সময় এসব মন্তব্য বিরাট উপকারী হবে। মনে রাখবেন, কোনো কোনো প্রোগ্রামের কোডের লাইন সংখ্যা শত শত থেকে হাজার হাজারে যেয়ে পৌঁছতে পারে!

১ম প্রোগ্রাম শেষ!

আপনি সফলভাবে প্রথম প্রোগ্রাম শেষ করেছেন। তবে আই.ডি.ই বন্ধ করার পূর্বে অবশ্যই একবার প্রোগ্রাম টেস্ট করে দেখতে হবে। নিচের চিত্র-৮-এ এই টেস্টের ফলাফল চিত্রিত হলো।



চিত্র-৮: ১ম প্রোগ্রাম ProjectBismillah পরিচালনা করে Click Me বোতামে ক্লিক করার ফলাফল।

৩য় পরিচ্ছেদ: ভিজুয়াল বেসিক.নেট ভাষার মৌলিক কাঠামো



ভিজুয়াল বেসিক.নেট : Subs & Functions

ডট.নেট প্রোগ্রামের কোড মূলত Sub [Subroutine] ও Function নিয়ে গঠিত। ১ম প্রোগ্রামে আমরা Sub ব্যবহার করেছি। এ দু'টো হলো প্রোগ্রাম কোডের মধ্যে একেকটি অংশ।

প্রত্যেক Sub শুরু হয় এভাবে [এর Syntax] :

```
[accessibility] Sub routine_name [paramaters]
    statements
End Sub
```

accessibility = Public, Protected, Friend, Protected Friend, or Private

Public -অর্থ এই সাবরুটিন সমগ্র প্রোগ্রামব্যাপী ব্যবহারযোগ্য।

Private -অর্থ যে মডিউলের ভেতর এটি আছে শুধু সেখানেই ব্যবহারযোগ্য।

আপাতত এ দু'টোই আমরা ব্যবহার করবো।

routine_name -যে কোনো নাম। নামটি ভিজুয়াল বেসিকের আইন মুতাবিক হতে হবে। যে কোনো অক্ষর বা আন্ডারস্কোর [-] দ্বারা নাম শুরু হবে। নামের মধ্যে কোনো স্পেস [শূন্যস্থান ক্যারাক্টার] থাকবে না।

paramaters - ঠিক ভেরিয়েবল ঘোষণার মতো। একটু পর কিছু উদাহরণ দ্বারা সব পরিষ্কার হবে।

প্রত্যেক Function শুরু হয় এভাবে [এর Syntax]:

```
[accessibility] Function function_name [paramaters] [As return_type ]
    statements
End Function
```

প্রতিটি আইটেম Sub ঘোষণার অনুরূপ। আর As return_type -এর অর্থ হলো কোন ধরনের তথ্য ফাংশন ফিরে দেবে।

মনে রাখবেন: প্রত্যেক ফাংশনের কাজ হলো কোনো তথ্য [ভ্যালু] ফিরে দেওয়া।

বিষয়টি স্পষ্টভাবে বুঝার জন্য নিচের কোড অংশ দেখুন (চিত্র-৯)। এই কোডিংয়ের উদাহরণ পরে আসছে।

চিত্র-৯:

Sub

এবং

Function

লেখা হয়েছে

ফর্মের ক্লাস

ব্লকের ভেতর।

```
Form1.vb [Design] Form1.vb [X]
ListBox1
SelectedIndexChanged
Equation: AreaBox = 2*(L*W + L*H + W*H) LD=length, HD=height, WD=width, SD=side width,
Equation: VolSpH = (4/3) * Pie * R^3 RS=radius of Sphere
Private Sub ListBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ListBox1.SelectedIndexChanged
    HideAll()
    TextBox6.Text = ListBox1.SelectedItem
    Dim WhatNeeded As String = TextBox6.Text
    If WhatNeeded = "" Then Exit Sub

    '***** Calculate Area and Volume of Cube *****
    If WhatNeeded = "Cube" Then
        Label3.Visible = True : TextBox1.Visible = True
        Label4.Visible = False : Label5.Visible = False : TextBox2.Visible = False : TextBox3.Visible = False
        Label3.Text = "Enter Cube Side length."
    End If

    '***** Calculate Area and Volume of Cylinder *****
    If WhatNeeded = "Cylinder" Then
        Label3.Visible = True : Label4.Visible = True : TextBox1.Visible = True : TextBox2.Visible = True
        Label5.Visible = False : Label5.Visible = False
        Label3.Text = "Enter Cylinder Radius."
        Label4.Text = "Enter Cylinder Height."
    End If

    '***** Calculate Area and Volume of Box *****
    If WhatNeeded = "Box" Then
        Label3.Visible = True : Label4.Visible = True : Label5.Visible = True
        TextBox1.Visible = True : TextBox2.Visible = True : TextBox3.Visible = True
        Label3.Text = "Enter Box Length."
        Label4.Text = "Enter Box Width."
        Label5.Text = "Enter Box Height."
    End If

    '***** Calculate Area and Volume of Sphere *****
    If WhatNeeded = "Sphere" Then
        Label3.Visible = True : TextBox1.Visible = True
        Label4.Visible = False : Label5.Visible = False : TextBox2.Visible = False : TextBox3.Visible = False
        Label3.Text = "Enter Sphere Radius."
    End If
End Sub

Function AreaSphere(Radi As Double) As Double
    Dim Pie As Double = 3.142
    Return 4 * Pie * Radi * Radi
End Function

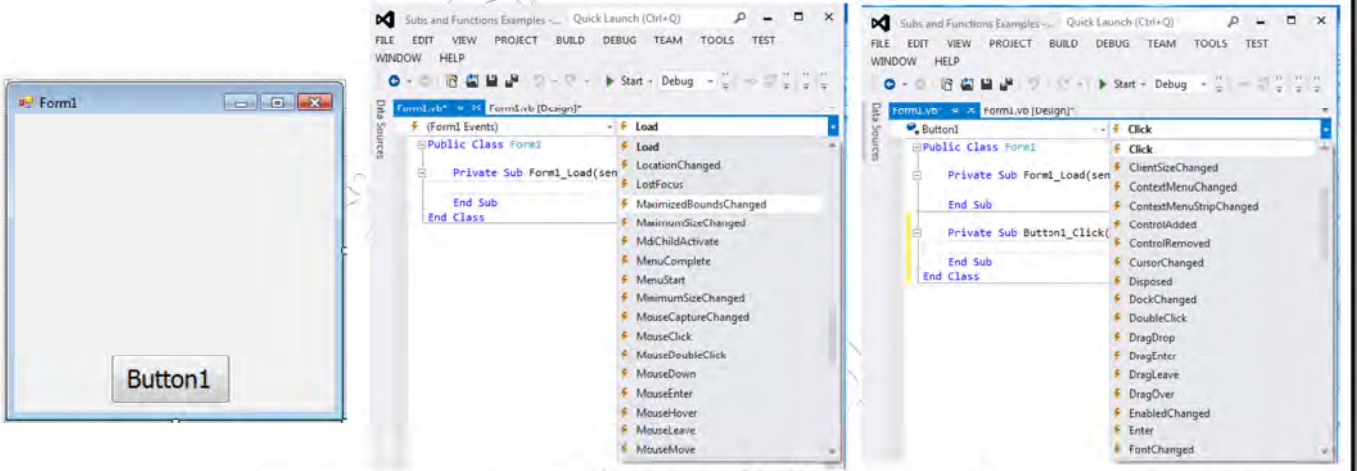
Function VolSphere(Radi As Double) As Double
    Dim Pie As Double = 3.142
    Return 4 / 3 * Pie * Radi * Radi * Radi
End Function
```

৩য় পরিচ্ছেদ: ভিজুয়াল বেসিক.নেট ভাষার মৌলিক কাঠামো

ভিজুয়াল বেসিক.নেট : Subs & Functions Examples

১ম উদাহরণ: Sub

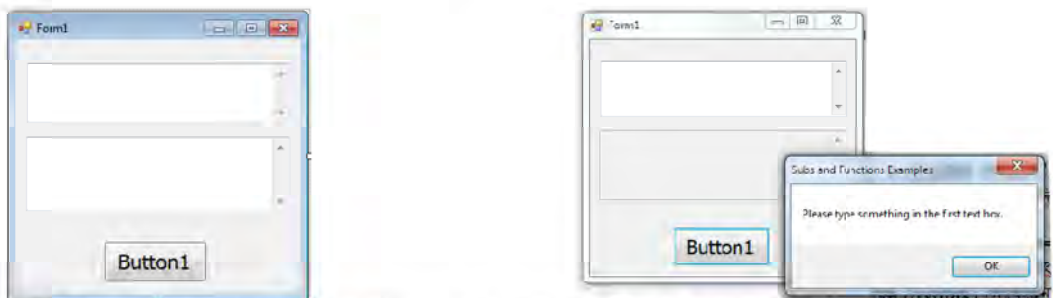
Sub এর উদাহরণ উপস্থাপনের পূর্বে আপনাকে একটি বিষয় খুব ভালো করে জানতে হবে। Form, Button, TextBox, Label, ListBox, ComboBox ইত্যাদি টুলবাক্সের যাবতীয় বস্তু [Objects] একাধিক event [ঘটনা] আছে। প্রোগ্রামার হিসাবে আপনার কাজ হলো এসব ঘটনাকে Sub -এর মধ্যে ধরে নিজের ইচ্ছে মতো কাজ করানো। নিচের চিত্রে [চিত্র-১০] ফর্ম ও বোতামের কিছু ঘটনা [events] দেখানো হয়েছে। এসব ইভেন্ট কোড উইন্ডোতে যেয়ে সিলেক্ট করলেই একেকটি Sub আই.ডি.ই বানিয়ে দেবে। আপনি এর ভেতর কোড লিখে ইভেন্টকে নিয়ন্ত্রণ ও কাজে লাগাবেন। আর এভাবে ইভেন্ট কাজে লাগানোর নামই প্রোগ্রামিং। আপনি যদি সফলভাবে এগুলো কাজে লাগাতে সক্ষম হন তাহলে আপনি হবেন একজন দক্ষ প্রোগ্রামার।



চিত্র-১০: (বায়ে) একটি Project। (মাঝে) প্রজেক্টের ফর্মের ইভেন্ট লিস্ট। (ডানে) প্রজেক্টের বাটন এর ইভেন্ট লিস্ট।

অধিকাংশ ক্ষেত্রে আপনি যেসব ইভেন্ট কাজে লাগাবেন তাহলো: (ফর্ম ও বোতামের ক্ষেত্রে) Click, MouseUp, MouseDown, MouseEnter, MouseLeave; (লিস্টবাক্স, কম্বোবাক্স, টেক্সটবাক্স ইত্যাদির ক্ষেত্রে) Click, MouseEnter, TextChanged, MouseLeave ইত্যাদি।

Sub শব্দের অর্থ Subroutine। ভিজুয়াল বেসিক.নেট -এ সাব -এর গুরুত্ব অপরিমিত। সুতরাং সাব কিভাবে কাজ করে তা আপনাকে জানতেই হবে। আপনি আগের পৃষ্ঠায় সাব লেখার নিয়ম (Syntax) দেখেছেন। পরের পৃষ্ঠায় সাব দু'টি উপরের চিত্র-১০ -এ চিত্রিত বোতামের Click event ও টেক্সট বাক্সের MouseLeave event সাব, যা আমরা লিখেছি দু'টি কাজ আঞ্জাম দিতে। উক্ত ফর্মে এবার দু'টি TextBox কন্ট্রোল যোগ করেছি। নিচের বায়ের চিত্রে (চিত্র-১১) দেখানো হয়েছে ফর্মের ডিজাইনটি। ডানের চিত্রে (চিত্র-১২) তুলে ধরেছি রান করার পরে বোতামে ক্লিক করার ফলাফল। আপনি যখন মেসেজ বাক্সের 'Ok' বোতামে ক্লিক করবেন তখনই ফিল্ডে কিছু লিখার জন্য প্রোগ্রাম অপেক্ষায় থাকবে।



চিত্র-১১: (বায়ে) ফর্মের ডিজাইন। (ডানে) প্রোগ্রাম চলাকালে বোতামে ক্লিক করলে যে মেসেজ আসবে তার দৃশ্য।

৩য় পরিচ্ছেদ: ভিজুয়াল বেসিক.নেট ভাষার মৌলিক কাঠামো

ভিজুয়াল বেসিক.নেট : Subs & Functions Examples

আগের পৃষ্ঠায় বর্ণিত উদাহরণের দু'টি Sub নিচে তুলে ধরেছি। এরপর (চিত্র-১৩) প্রোগ্রাম চলাকালে প্রথম টেক্সট বাক্সের লেখা বাক্স থেকে মাউস সরে নিয়ে আসার পর দ্বিতীয় টেক্সট বাক্সে লেখা কপি হয়েছে দেখা যাচ্ছে।

Public Class Form1

'This sub is the button click event code

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

MsgBox("Please type something in the first text box.") 'call messageBox to display

'instruction

End Sub

'Catching MouseLeave event and doing something

Private Sub TextBox1_MouseLeave(sender As Object, e As EventArgs) Handles TextBox1.MouseLeave

If TextBox1.Text = "" Then Exit Sub 'checks if there is text in the first field. if not program ends.

TextBox2.Text = TextBox1.Text 'if there is text, this line executed, copies

'text into 2nd field.

End Sub

End Class

উক্ত সাব দু'টো কিভাবে কাজ করে: প্রোগ্রাম রান করলেই পুরো ফর্মটি প্রদর্শিত হবে। আপনি যদি বোতামে ক্লিক করেন তাহলে প্রথম সাবটি কাজ শুরু করবে। এই সাবে আমরা নির্দেশ দিয়েছি: একটি মেসেজ বাক্স প্রদর্শন করতে যেখানে লিখা থাকবে, "Please type something in the first text box." কথাটি। 'Ok' ক্লিক করার পর আগের অবস্থায় ফিরে যাবে। এবার আপনি যদি কিছু লিখেন এবং মাউসটি বাক্স থেকে বের করেন তাহলে সাথে সাথে লেখাটুকু এমনিতেই দ্বিতীয় বাক্সে চলে যাবে। এই কাজটি সম্ভব হয় ১ম টেক্সট বাক্সের মাউসলীড সাব সক্রিয় হওয়ার কারণে। এ সাবে আমরা নির্দেশ দিয়েছি: প্রথমে পরীক্ষা করো, বাক্সে কিছু লেখা হলো কি না? যদি না হয়- সাব থেকে বের হও। যদি লেখা থাকে তাহলে দ্বিতীয় লাইনের নির্দেশ পালন করো। অর্থাৎ ১ম বাক্সের যাবতীয় লেখা ২য় বাক্সে লিখ। এরপর সাব থেকে বের হও।



চিত্র-১৩: প্রোগ্রাম Run করে প্রথম টেক্সট বাক্সের লিখা স্বয়ংক্রিয়ভাবে দ্বিতীয়টি লেখা হওয়ার দৃশ্য।

২য় উদাহরণ: Function

উপরের ১ম উদাহরণে আমরা Sub-এর ব্যবহার দেখেছি। এবার একটি নিজের তৈরি ফাংশন (Function) উপস্থাপন করছি। ভিজুয়াল বেসিকে Function সর্বদাই কোনো এক বা একাধিক ভ্যালু (value) রিটার্ন করে। এটা হতে পারে প্রাইভেট বা পাবলিক বা আরো কোনো ধরনের। ইতোমধ্যে আপনি ফাংশনের সিন্টেক্স (Syntax) দেখেছেন। যাক, উক্ত প্রজেক্টে এখন আমরা একটি ফাংশন যোগ করবো।

** মনে রাখবেন: ফাংশন ব্লক ফর্মের ক্লাসের ভেতর যে কোনো স্থানে লিখা যায়। এটাকে যে কোনো সাব বা অপর ফাংশন থেকে কল (call) করা হয়ে থাকে। ফাংশন (এবং সাবকে) বার বার কল করা সম্ভব। আর সর্বদাই ফাংশন থেকে ফলাফল (ভ্যালু) আসবে। ভ্যালুর ধরন (type) যে কোনোটি হতে পারে, যেমন: text, number, etc. **

উদ্দেশ্য: আমাদের ফাংশনের কাজ হলো, আপনি মোট ক'টি অক্ষর টাইপ করেছেন তা একটি মেসেজ বাক্সে দেখানো। সুতরাং ফাংশনের রিটার্ন ভ্যালু হবে একটি ইন্টজার (আন্ত) সংখ্যা। পরের পৃষ্ঠায় এ সম্পর্কে বিস্তারিত দেখুন।

ওয় পরিচ্ছেদ: ভিজুয়াল বেসিক.নেট ভাষার মৌলিক কাঠামো

ভিজুয়াল বেসিক.নেট : Subs & Functions Examples

আমাদের ফর্ম ডিজাইনে কোনো পরিবর্তনের প্রয়োজন নেই। আমাদের কাজ হলো ফাংশন ব্লকটি ফর্মের ক্লাসের ভেতর লেখা। এরপর এটি অন্য কোনো কোড ব্লক থেকে কল করা। এই যা। নিচে ফাংশন কোডসহ পুরো কোড তুলে ধরেছি।

Public Class Form1

'This sub is the button click event code

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

MsgBox("Please type something in the first text box.") 'call messageBox to display

'instruction

End Sub

'Caching MouseLeave event and doing something

Private Sub TextBox1_MouseLeave(sender As Object, e As EventArgs) Handles TextBox1.MouseLeave

If TextBox1.Text = "" Then Exit-Sub 'checks if there is text in the first field

TextBox2.Text = TextBox1.Text 'if there is text this line executed. copies

' text into 2nd field.

'function call from message box

MsgBox("You have typed: " & NumOfChar(TextBox1.Text) & " letters.")

End Sub

'function returns number of characters in TextBox1

Function NumOfChar(Supply As String) As Integer

If TextBox1.Text = "" Then 'in case there aren't any characters in the textBox1

'although it is checked at the beginning of the sub. This If --- End If block is for extra solidarity.

Return 0 'empty textBox means no character.

End If

Return Len(TextBox1.Text) 'counts character and returns integer

End Function

End Class

প্রোগ্রামে ফাংশনের ব্যবহার: আমাদের সৃষ্ট NumOfChar (Supply) ফাংশন কিভাবে ব্যবহৃত হয়েছে, তা একটু লক্ষ্য করুন। টেক্সটবক্স১ এর 'TextBox1_MouseLeave' ইভেন্ট সাব থেকে এটি কল করেছে। ফলাফল একটি সংখ্যা যা মেসেজ বাক্সে লিখা হবে। অন্যকথায় মেসেজ বাক্সের ভেতর থেকেই আমরা কলটি করেছি। চাইলে এভাবেও হতো:

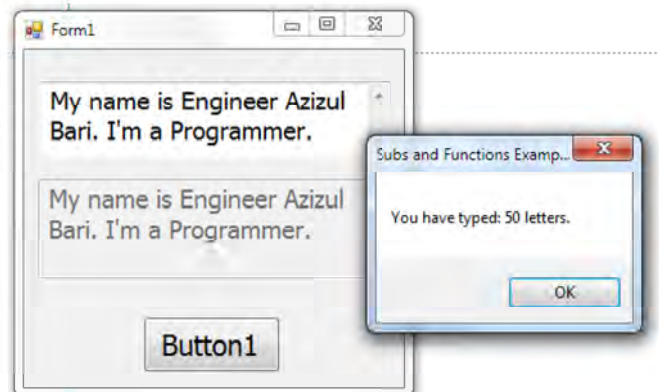
Dim NumOfLetters as Integer

NumOfLetters = NumOfChar(TextBox1.Text) 'Function call

MsgBox("You have typed: " & NumOfLetters & " letters.")

উক্ত কোড দ্বারা প্রথমে আমরা NumOfLetters নামক একটি ভেরিয়েবলে বাক্সের অক্ষর সংখ্যা ফাংশন কল করে সংরক্ষণ করেছি। এরপর এটা মেসেজ বাক্স দ্বারা প্রদর্শিত হয়েছে। তবে এ ক্ষেত্রে দু'টি অতিরিক্ত লাইন কোডিং হয়েছে। যাক, নিচে ফাংশনের কাজ (চিত্র-১৪) চিত্রিত হয়েছে।

চিত্র-১৪:
NumOfChar
(Supply)
ফাংশন -এর
ব্যবহার ও
ফলাফল।



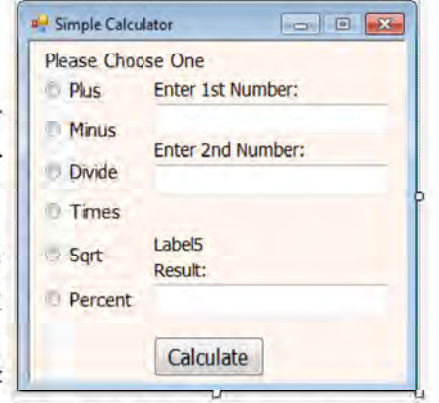
ওয় পরিচ্ছেদ: ভিজুয়াল বেসিক.নেট ভাষার মৌলিক কাঠামো

ভিজুয়াল বেসিক.নেট : Subs & Functions Examples

ওয় উদাহরণ: Sub & Function

এই উদাহরণে আমরা সত্যিকার অর্থে ব্যবহারযোগ্য একটি ক্যালকুলেটর তৈরি করতে যাচ্ছি। যে কোনো পিসিতে এটি ব্যবহার করা যাবে। প্রথমে আমরা ফর্মের ডিজাইন নিয়ে কিছু আলোচনা করবো। নিচের চিত্রে (চিত্র -১৫) এটি দেখানো হয়েছে। নিম্নে বর্ণিত পদ্ধতিতে প্রজেক্ট তৈরি করে ফর্ম ডিজাইন করুন।

1. একটি নতুন প্রজেক্ট তৈরি করুন। এর নামকরণ করুন SimpleCal
2. ফর্মের সাইজ প্রোপার্টি হবে: ৩১৫ x ৩১০। Name প্রোপার্টি হবে Simple Calculator। ফর্মের FormBorderStyle প্রোপার্টি হবে Fixed3D এবং MaximizeBox প্রোপার্টি হবে False।
3. ফর্মে ৫টি লেবেল থাকবে: label1 এর Text প্রোপার্টি হবে Please Choose One। label2 এর Text প্রোপার্টি হবে Enter 1st Number। label3 এর Text প্রোপার্টি হবে Enter 2nd Number। label4 এর Text প্রোপার্টি হবে Result:। label5 এর Text প্রোপার্টি হবে Label5- এটা দেওয়াই থাকবে।
4. ফর্মে থাকবে ৩টি টেক্সট বক্স: TextBox1, TextBox2 ও TextBox3। সবগুলোর Font Size প্রোপার্টি হবে 11। শুধু TextBox3 -এর ReadOnly প্রোপার্টি হবে True। আমরা চাইনা এই বাক্সে ব্যবহারকারী কিছু লিখবেন।
5. ফর্মে মোট ৬টি রেডিও বাটন থাকবে: এগুলোর ক্যাপশন নাম (Text property) হবে যথাক্রমে, Plus, Minus, Divide, Times, Sqrt এবং Percent। এদের (Name) প্রোপার্টি হবে যথাক্রমে, RDplus, RDminus, RDdivide, RDtimes, RDsqrt এবং RDpercent। এই প্রোপার্টি সঠিকভাবে থাকা চাই। এ নামগুলো কোডে ব্যবহৃত হবে।
6. সবশেষে ফর্মে একটি বোতাম থাকবে। এর Font Size প্রোপার্টি হবে 12 এবং Text প্রোপার্টি হবে Calculate। চাইলে ফর্মের BackColor প্রোপার্টি বদলাতে পারেন। আমি হালকা হলুদ (SeaShell) দিয়েছি।



চিত্র -১৫: সিম্পুল ক্যালকুলেটর প্রজেক্টের ফর্ম

ডিজাইন শেষ হওয়ার পর একবার Start মেনুতে ক্লিক করে রান করে দেখুন। লক্ষ করুন, ফর্ম বড় কিংবা ছোট করা যাবে না। আমরা এটাই চাই। ক্লোজ বাক্সে ক্লিক করে বন্ধ করুন।

কোডিং

এ উদাহরণে মূল কোড লিখিত হবে Calculate বোতামের Click ইভেন্টে। তবে ফর্মের Load ইভেন্টে একটি ছোট কোড লেখার প্রয়োজন। আমরা চাই রেডিও বোতামের Default একটি চয়েজ থাকবে। এটি হবে Plus নামক রেডিও বোতামটি (অর্থাৎ প্রথমটি)। সুতরাং ফর্মের লোড কোড হবে নিম্নরূপ:

```
' default value 'Plus' -radio button1 chosen
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    RDplus.Checked = True
End Sub
```

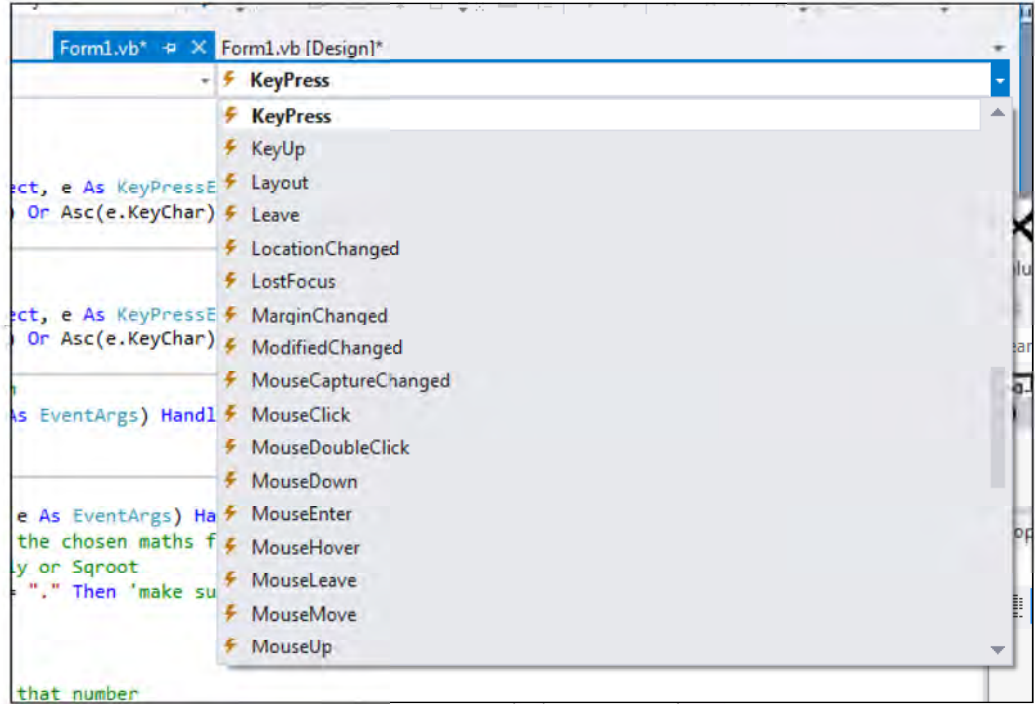
Form1.vb [design] ক্লিক করে ফর্মের যে কোনো স্থানে [কোনো বস্তুতে নয়] ডবল ক্লিক করলেই উক্ত সাবটি আসবে। এখানে শুধু মাঝের লাইনটি লিখবেন। এই লাইন দ্বারা বুঝানো হচ্ছে: ফর্মের Plus Radio button চেক করা থাকবে। এর দ্বারা এটাও বুঝাচ্ছে যে, ব্যবহারকারী উভয় টেক্সট বাক্সে সংখ্যা লিখে Calculate বোতামে ক্লিক করলে উভয়টির যোগফল আসবে।

ওয় পরিচ্ছেদ: ভিজুয়াল বেসিক.নেট ভাষার মৌলিক কাঠামো

ভিজুয়াল বেসিক.নেট : Subs & Functions Examples

Calculate বোতামের Click ইভেন্টের কোডিংয়ে যাওয়ার পূর্বে আমরা আরো দু'টি ছোট কোডিং করবো। এগুলো অনুরূপ এবং লিখিত হবে TextBox1 ও TextBox2 -এর KeyPress নামক ইভেন্টে। এই সাবটি পেতে হলে প্রথমে টেক্সটবক্সে ডবল ক্লিক করুন। এতে টেক্সটবক্সের TextChanged নামক ইভেন্টটি আসবে। এটা আমাদের প্রয়োজন নেই। তবে উপরে ডানদিকের মেনু থেকে চয়েজ করে নিতে হবে টেক্সটবক্সের KeyPress নামক ইভেন্টটি (নিচের চিত্রটি দেখুন)। নিচের সাবটি আসবে। এই সাবের মধ্যেই আমরা এক লাইন কোড লিখেছি।

চিত্র -১৬: কিভাবে
টেক্সটবক্সের
KeyPress ইভেন্ট সাব
বের করবেন?



```

'Make sure only numbers can be entered
Private Sub TextBox1_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
    TextBox1.KeyPress
        e.Handled = Not (Char.IsDigit(e.KeyChar) Or Asc(e.KeyChar) = 8 Or ((e.KeyChar
        = ".") And (sender.Text.IndexOf(".") = -1)))
    End Sub

```

ঠিক একই কোড যাবে TextBox2 এর KeyPress ইভেন্ট সাবেও:

```

'Make sure only numbers can be entered
Private Sub TextBox2_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
    TextBox2.KeyPress
        e.Handled = Not (Char.IsDigit(e.KeyChar) Or Asc(e.KeyChar) = 8 Or ((e.KeyChar
        = ".") And (sender.Text.IndexOf(".") = -1)))
    End Sub

```

উপরের কোড লাইনটি ভালো করে শিখে নিন। এই কোড দ্বারা যে কোনো টেক্সটবক্সে শুধুমাত্র সংখ্যা (দশমিক একটিমাত্র বিন্দুসহ '.') এবং Delete ছাড়া অন্যকিছু যাতে কেউ লিখতে পারেন না তা নিশ্চিত করে। প্রয়োজনে কোডটি নোট বইয়ে লিখে রাখুন।

যাক এ পর্যন্ত আমরা যা করেছি তা দেখার জন্য একবার প্রোগ্রাম রান করে দেখুন। উভয় টেক্সটবক্সে সংখ্যা লিখুন এবং সংখ্যা ছাড়া অন্য কিছু লিখার চেষ্টা করুন। এবার আমরা আসল কোড নিয়ে বিস্তারিত আলোচনা করবো।

ওয় পরিচ্ছেদ: ভিজুয়াল বেসিক.নেট ভাষার মৌলিক কাঠামো

ভিজুয়াল বেসিক.নেট : Subs & Functions Examples

Calculate বোতামের কোড

আমরা ইতোমধ্যে ফর্মের লোড ইভেন্ট ও সংখ্যা লেখার কীপ্রেস ইভেন্ট কোড নিয়ে আলোচনা করেছি। এই কোডগুলো দ্বারা প্রথমত Plus রেডিও বোতাম চয়েজ করা অবস্থা ও টেক্সটবক্স দু'টোয় শুধুমাত্র সংখ্যা লিখা সম্ভব- তা নিশ্চিত করেছি। এবার আমরা ক্যালকুলেট বোতামের কোড নিয়ে আলোচনায় যেতে পারি। নীচে প্রথমে বোতামের কোডগুলো তুলে ধরলাম। এরপর প্রত্যেক লাইনের ওপর ব্যাখ্যা দেবো।

```
'calculation button
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim ActWant As String = WhichCal() 'get the chosen maths fuction value
    ' ActWant= Add, Minus, Division, Multiply or Sqrt
    If TextBox1.Text = "" Or TextBox1.Text = "." Then
        Exit Sub 'make sure there is 'number in the first fld
    End If
    'first fld has a number
    'check if the user wants to get sqrt of that number
    If TextBox1.Text <> "" And ActWant = "Sqrt" Then
        TextBox2.Text = "" ' Note: if second textBox also has number it is
        'ignored and wiped
        Label5.Text = "√" & TextBox1.Text
        TextBox3.Text = Math.Sqrt(Val(TextBox1.Text)) 'calculate square root, display
        'and exit
        Exit Sub
    End If
    'textBox1 has a number and chosen radio button is not sqrt
    'check if textBot2 has a valid number
    If TextBox2.Text = "" Or TextBox2.Text = "." Then 'make sure there is number in
        'the first fld
        Exit Sub
    End If
    'check if the user trying to divide by 0
    If Val(TextBox2.Text) = 0 And ActWant = "Division" Then
        TextBox3.Text = "Division by 0 not valid!"
        Exit Sub
    End If
    'both textBox have valid numbers- calculate and display results
    If TextBox1.Text <> "" Or TextBox2.Text <> "" Then 'check again
        If ActWant = "Add" Then
            Label5.Text = TextBox1.Text & " + " & TextBox2.Text 'display info
            TextBox3.Text = Val(TextBox1.Text) + Val(TextBox2.Text) 'add and show
            Exit Sub
        End If
        If ActWant = "Minus" Then
            Label5.Text = TextBox1.Text & " - " & TextBox2.Text 'display info
            TextBox3.Text = Val(TextBox1.Text) - Val(TextBox2.Text) 'minus and show
            Exit Sub
        End If
        If ActWant = "Division" Then
            Label5.Text = TextBox1.Text & " ÷ " & TextBox2.Text 'display info
            TextBox3.Text = Val(TextBox1.Text) / Val(TextBox2.Text) 'divide and show
            Exit Sub
        End If
    End If
```

৩য় পরিচ্ছেদ: ভিজুয়াল বেসিক.নেট ভাষার মৌলিক কাঠামো

ভিজুয়াল বেসিক.নেট : Subs & Functions Examples

```

If ActWant = "Multiply" Then
    Label5.Text = TextBox1.Text & " x " & TextBox2.Text 'display info
    TextBox3.Text = Val(TextBox1.Text) * Val(TextBox2.Text) 'multiply and show
    Exit Sub
End If
If ActWant = "Percent" Then
    Label5.Text = TextBox2.Text & "% of " & TextBox1.Text 'display info
    TextBox3.Text = Val(TextBox1.Text) * Val(TextBox2.Text) / 100 'get percent
    'and show
    Exit Sub
End If
End If
End Sub
    
```

উপরের কোডের ব্যাখ্যা

প্রথম লাইন: `Dim ActWant As String = WhichCal()` দ্বারা আমরা ActWant নামক একটি ভেরিয়েবল ঘোষণা করছি। এটার মধ্যে একটি শব্দ থাকবে যা নির্ভর করবে ব্যবহারকারী কোন্ রেডিও বোতাম চয়েজ করেছেন তার ওপর। লক্ষ করুন একই লাইনে ভেরিয়েবলের ভ্যালু নির্ধারণ করা হয়েছে WhichCal() নামক ফাংশন কল করে। এই ফাংশন আমরা আলাদাভাবে ডিফাইন করেছি। এর ওপর আলোচনা একটু পরে আসছে। এখানে শুধু এটুকু জানতে হবে যে, এটা রিটার্ন করে ব্যবহারকারী কোন্ রেডিও বোতাম চয়েজ করেছেন তার একটি নাম। সুতরাং প্রথম লাইন থেকে আমরা জানলাম ব্যবহারকারী কোন্ ক্যালকুলেশন করবেন। এবার দেখে নেই WhichCal() ফাংশনের কোডটি।

```

Function WhichCal() 'Returns values according to which radio button clicked
    If RDplus.Checked = True Then
        Return "Add"
    End If
    If RDminus.Checked = True Then
        Return "Minus"
    End If
    If RDdivide.Checked = True Then
        Return "Division"
    End If
    If RDtimes.Checked = True Then
        Return "Multiply"
    End If
    If RDsqrt.Checked = True Then
        Return "Sqroot"
    End If
    If RDpercent.Checked = True Then
        Return "Percent"
    End If
    Return "Add"
End Function
    
```

লক্ষ করুন: এই ফাংশনের কোনো প্যারামিটার নেই। এটার কাজ হলো ব্যবহারকারী কোন্ রেডিও বোতাম ক্লিক করে রেখেছেন তা পরীক্ষা করে একটি নাম [অর্থাৎ, 'Add', 'Minus', 'Division', 'Multiply', 'Sqroot' কিংবা 'Percent'] ফেরৎ পাঠানো। ফাংশনের Return কিওয়ার্ড এ কাজটি করে। এই কিওয়ার্ড পরে ফাংশনের কাজ শেষ হয়ে যায়। পরের লাইনে আর যাবে না। মনে করুন, আপনি চয়েজ করলেন Times নামক রেডিও বোতামটি- অর্থাৎ পূরণ অঙ্ক করবেন। এই ফাংশন একে একে চেক করে দেখবে কোন্ রেডিও বোতাম চেক করা অবস্থায় আছে। যেতেহু RDtimes নামক বোতামটি চয়েজ করা আছে- তাই এখানে এসে ফাংশন রিটার্ন করবে Multiply ভ্যালু। অন্যকথায়, এক্ষেত্রে ActWant এর মান হবে Multiply।

৪র্থ পরিচ্ছেদ: ভিজুয়াল বেসিক.নেট ভাষার Variable Declaration

ভিজুয়াল বেসিক.নেট : Variable & Data Type

আমরা ইতোমধ্যে ভিজুয়াল বেসিক.নেট এর কয়েকটি উদাহরণ দেখেছি। এই পরিচ্ছেদেও আরো কিছু উদাহরণ দেখবো। তবে এসব উদাহরণের উদ্দেশ্য হবে ভাষার Variable Declaration এবং Data type সম্পর্কে অবগত হওয়া।

Variable কি?

যে কোনো প্রোগ্রামিং ভাষায় পরিবর্তনযোগ্য কিছু তথ্যের প্রয়োজন পড়ে। যে উপায়ে এটা সম্ভব তাহলো কম্পিউটারের মেমোরিতে তথ্যের ধরন চিহ্নিত করে বিশেষ নাম দ্বারা জায়গা সংরক্ষণ করা। আর প্রয়োজনে প্রোগ্রাম থেকে ঐ তথ্যটুকু এনে এর উপর কাজ, রদবদল, হিসাব-নিকাশ, পরিবর্তন ইত্যাদি করে পুনরায় সংরক্ষণ করা। এছাড়া দৃশ্যমান বস্তু যেমন টেক্সট বাক্স, লেবেল, লিস্ট বাক্স ইত্যাদিতে প্রদর্শন করা। মোটকথা ভেরিয়েবল ছাড়া কোনো কম্পিউটার প্রোগ্রাম লিখা প্রায় অসম্ভব। সুতরাং এটা কিভাবে ডিক্লার করার করতে হয় এবং কোন্ উপায়ে কাজে লাগানো যায় তা জানা আমাদের জন্য একান্ত জরুরী।

কিভাবে ভেরিয়েবল ডিক্লারেশন করবেন?

প্রত্যেক ভেরিয়েবল নল্লিখিত উপায়ে ডিক্লার করার করতে হবে:

১. স্কোপ বা ব্যবহারের ব্যাপ্তি: ফর্মের কোথায় এটা ডিক্লার করার করছেন তা-ই নির্ধারণ করে তার স্কোপ। ফর্মের ক্লাস ডিক্লারেশনের পরই (অর্থাৎ প্রথম লাইন বাদে) ডিক্লার করার করলে ক্লাসের ভেতরস্থ সকল প্রসিডোর, সাব, ফাংশন ইত্যাদিতে এটা কাজে লাগানো যায়। অপরদিকে যে কোনো প্রসিডোর, সাব ও ফাংশনের ভেতর ডিক্লার করার করলে শুধু ঐ সাবের ভেতরই এটা ব্যবহারযোগ্য হবে। নিচের উদাহরণ দু'টো দেখুন।

উদাহরণ ১. সর্বত্র ব্যবহারযোগ্য ভেরিয়েবল।

Public Class Form1

```
Dim intNum As Integer 'intNum scope throughout rest of the program
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    intNum = 20 'Here intNum = 20
End Sub
Private Sub TextBox1_MouseLeave(sender As Object, e As EventArgs) Handles TextBox1.MouseLeave
    intNum = 50 'Here intNum = 50
End Sub
Function NumOfChar()
    intNum = Len(TextBox1.Text) 'Here intNum = Number of characters in TextBox 1
    Return intNum
End Function
End Class
```

উদাহরণ ২. শুধুমাত্র সাব বা ফাংশনের ভেতর ব্যবহারযোগ্য।

Public Class Form1

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    dim EntText as String = TextBox1.Text 'only available in this sub
End Sub
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    MsgBox(EntText) 'will not work -EntText is not declared error comes- because here it isn't useful.
End Sub
End Class
```

৪র্থ পরিচ্ছেদ: ভিজুয়াল বেসিক.নেট ভাষার **Variable Declaration**

ভিজুয়াল বেসিক.নেট : **Variable & Data Type**

Data Type

কম্পিউটারে বুঝার জন্য যে কোনো তথ্যের ধরন বা টাইপ কি তা বলে দিতে হবে। ভেরিয়েবল ঘোষণার সময় এর Type-ও লিখে দিতে হয়। যেমন:

Dim MyVariable as String 'String is the type

Dim MyVariable as Integer 'Integer is the type

Dim MyVariable as Single 'Single (decimal number) is the type

এই গ্রন্থে ব্যবহৃত বিভিন্ন Type

1. Boolean - True or False
2. Char (single Character) - 0 through 65535 (unsigned)
3. Date - 0:00:00 (midnight) on January 1, 0001 through 11:59:59 PM on December 31, 9999
4. Double (double precision floating point number)
5. Single (single precision floating point number)
6. Integer - -2,147,483,648 through 2,147,483,647 (signed)
7. Long (long Integer)
8. Object - Any type can be stored in a variable of type Object
9. Short (short integer) - -32,768 through 32,767 (signed)
10. String - 0 to approximately 2 billion Unicode characters

এবার আমরা উপরোক্ত প্রতিটি টাইপ ডিক্লারেশন ও ব্যবহার পদ্ধতির উদাহরণ দেখবো।

1. Boolean

Dim RadButOn As Boolean

' Check to see if RadioButton1 is ON or OFF.

If RadioButton1.Checked = True Then

RadButOn = True

Else

RadButOn = False

End If

2. Char

'Normally we don't need this type. It is used to hold a single character only.

Dim MyChar as Char

MyChar = "A"

3. Date

Dim date1 As Date = Date.Now

'gives local date and time: ex. 6/21/2016 1:34:14

Dim date2 As Date = Date.UtcNow

'gives short date and coordinated universal time (GMT time without daylight change: ex. 6/21/2016 7:34:14)

Dim date3 As Date = Date.Today

'gives just the short date: ex. 6/21/2016

৪র্থ পরিচ্ছেদ: ভিজুয়াল বেসিক.নেট ভাষার **Variable Declaration**

ভিজুয়াল বেসিক.নেট : **Variable & Data Type**

4. Double

Dim DecOne as Double = 23.09898731

Dim DecTwo as Double = 5.98987654

MsgBox(DecOne * DecTwo) 'gives 138.360082185927

5. Single

Dim DecOne as Single = 23.09898731

Dim DecTwo as Single = 5.98987654

MsgBox(DecOne * DecTwo) 'gives 138.3601

উপরোক্ত Double এবং Single টাইপের মধ্যে পার্থক্য লক্ষ করুন। প্রথমটি দ্বিতীয়টি থেকে অধিক সঠিক।

6. Integer

Dim MyNum as Integer

লক্ষ করুন: ছোট আস্ত সংখ্যা নিয়ে কাজ করতে ইন্টজার টাইপ ব্যবহার করা উত্তম।

7. Long

Dim MyNum as Long

লক্ষ করুন: বড় আস্ত সংখ্যা নিয়ে কাজ করতে লং টাইপ ব্যবহার করা উত্তম।

8. Object

'Any type can be stored in an Object Data Type

Dim v As Object

v = "17"

' v contains the 2-character String value "17".

v = v - 15

' v now contains the Integer value 2.

v = "H" & v

' v now contains the 2-character String value "H2"

9. Short

Dim ANum as Short 'Use short data type if you work with small integer to save memory

10. String

Dim j As String = "Joe said ""Hello"" to me."

Dim h As String = "Hello"

' The following messages all display the same thing:

' "Joe said "Hello" to me."

MsgBox(j)

MsgBox("Joe said " & """" & h & """" & " to me.")

MsgBox("Joe said """" & h & """" to me.")

সাধারণত আমরা Single, Long এবং Short Data type ব্যবহার করবো।

৪র্থ পরিচ্ছেদ: ভিজুয়াল বেসিক.নেট ভাষার Variable Declaration

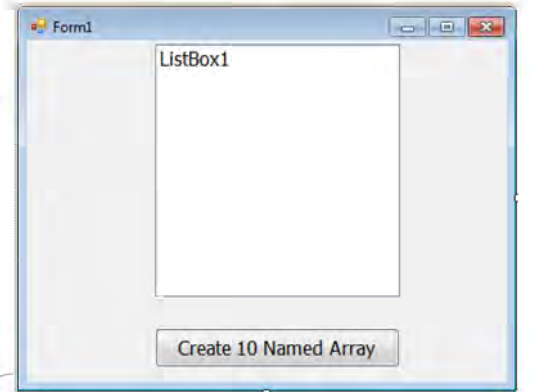
ভিজুয়াল বেসিক.নেট : Variable & Data Type

সংরক্ষিত নামগুলো এক সঙ্গে দেখতে চাইলে আমরা ব্যবহার করতে পারি একটি লিস্টবক্স। নিম্নের কোড অংশ উপরের কোডের নিচে লিখলেই হবে।

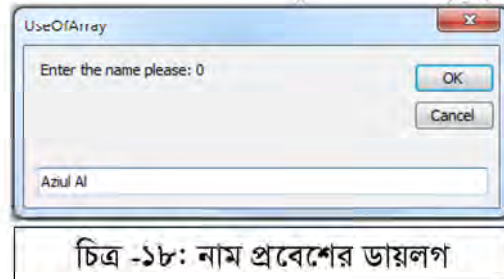
```
ListBox1.Items.Clear() ' display the result
For I = 0 To 7
    ListBox1.Items.Add(My10Names(I))
Next
```

Test 1: নতুন একটি প্রজেক্ট তৈরি করে এতে একটি লিস্টবক্স ও একটি বোতাম দিন। বোতামের নাম দিন 'Create 10 Names Array'। বোতামের ক্লিক ইভেন্টে উপরের কোড লিখে রান করুন। লিস্ট বাক্সে সব নাম এসছে কি? সমস্যা হলে নিচের ছবিগুলো দেখুন।

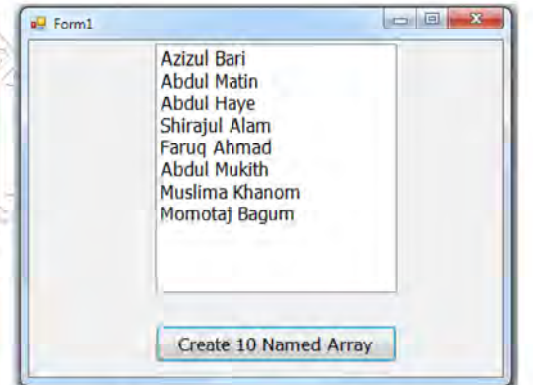
```
Public Class Form1
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Dim My10Names(9) As String 'define thearray
    Dim C As Integer ' define a counter
    ' fill the array
    For C = 0 To 9
        My10Names(C) = InputBox("Enter the name please: " & C)
    Next
    ListBox1.Items.Clear() ' display the result
    For I = 0 To 7
        ListBox1.Items.Add(My10Names(I))
    Next
End Sub
End Class
```



চিত্র -১৭: ফর্ম ডিজাইন



চিত্র -১৮: নাম প্রবেশের ডায়ালগ



চিত্র -১৯: রানের পর লিস্ট

মাইক্রোসফট ভিজুয়াল বেসিক.কম উপরোক্ত 'Array' তৈরির সুযোগ দিয়ে অনেক তথ্য সংরক্ষণ ও ব্যবহারের ব্যবস্থা করে দিয়েছে। তবে এখানেই শেষ নয়। এ্যারে ছাড়াও আরো একটি গুরুত্বপূর্ণ তথ্য সংরক্ষণ ও ব্যবহার পদ্ধতি আন্স্কিার করেছে। এটা এ্যারে থেকেও উন্নতমানের এবং সহজ। এই ব্যবস্থার নাম collection। আমরা এখন এ সম্পর্কে বিস্তারিত আলোচনায় যাচ্ছি।

৪র্থ পরিচ্ছেদ: ভিজুয়াল বেসিক.নেট ভাষার **Variable Declaration**

ভিজুয়াল বেসিক.নেট : **Variable & Data Type**

Collection Declaration & it's use -কালেকশন ঘোষণা ও এর ব্যবহার

Collection অনেকটা এ্যারের মতোই কাজ করে। তবে, কালেকশনের মধ্যে একাধিক মান বা ভেরিয়েবল সংরক্ষণ ও ব্যবহার করা যায় একই সঙ্গে। নিচে এ্যারে ও কালেকশনের মধ্যে মূল ক'টি পার্থক্য তুলে ধরেছি।

ক. এ্যারের ইনডেক্স শুরু হয় ০ থেকে। কিন্তু কালেকশনের ইনডেক্স শুরু হয় ১ থেকে। যেমন:

Dim AnArray(3) - তে চারটি ইলিমেন্ট বা তথ্য থাকবে: AnArray(0), AnArray(1), AnArray(2) ও AnArray(3)। অপরদিকে:

```
Dim MyCollection as New Collection
```

```
MyCollection.add (1, "Sharif")
```

```
MyCollection.add (2, "Nasim")
```

```
MyCollection.add (3, "Abdullah")
```

```
MyCollection.add (4, "Khaled")
```

উপরের কালেকশনে চারটি ইলিমেন্ট সংরক্ষিত হবে। আমরা এদেরকে ব্যবহার করবো এভাবে রেফারেন্স করে:

```
MyCollection(1), MyCollection(2), MyCollection(3) ও MyCollection(4)।
```

খ. আরেক পার্থক্য হলো ভেরিয়েবল টাইপ। এ্যারের মধ্যে সকল ইলিমেন্ট হতে হবে একই ধরনের। যেমন:

```
Dim MyArray (0 to 5) as integer -এখানে ০ থেকে ৫ পর্যন্ত ৬টি আস্ত সংখ্যা ঘোষণা করা হয়েছে।
```

```
Dim MyArray ("Aziz", "Matin", "Zakir") as string -এখানে তিনটি নাম যাদের সবার ভেরিয়েবল টাইপ হলো স্ট্রিং ঘোষণা করা হয়েছে। অপরদিকে কালেকশনের ইলিমেন্টগুলো বিভিন্ন ধরনের হতে পারে। যেমন:
```

```
Dim MyCollection as New Collection
```

```
MyCollection.add ("Sharif")
```

```
MyCollection.add (2)
```

```
MyCollection.add ("Hello!")
```

```
MyCollection.add ("01732562315")
```

গ. আরেক পার্থক্য হলো কালেকশনের ইলিমেন্ট সরাসরি যুক্ত ও সরানো যায়। অপরদিকে এ্যারে ফিক্স। যেমনে তেমনে ভ্যালু পরিবর্তন করা যায় না।

ঘ. আরো একটি পার্থক্য হলো, এ্যারে ইলিমেন্ট সহজেই পরিবর্তন বা অভাররাইট করা যায়। যেমন:

```
Dim MyArray (2)
```

```
MyArray (0) = 5
```

```
MyArray (0) = 7
```

কালেকশনের সময় এটা করা যাবে না। তবে অন্যভাবে করার ব্যবস্থা আছে। পরে আমরা সে সম্পর্কে আলোচনা করবো।

কালেকশনের কয়েকটি মেথড আছে যেমন:

```
Dim C As New Collection 'Declaration
```

```
C.Add (1, "Aziz") 'adding element
```

```
C.Add (2, "Hello!")
```

```
C.Remove(1) 'remove element in index 1
```

```
I = C.Count 'get number of elements in C as integer
```

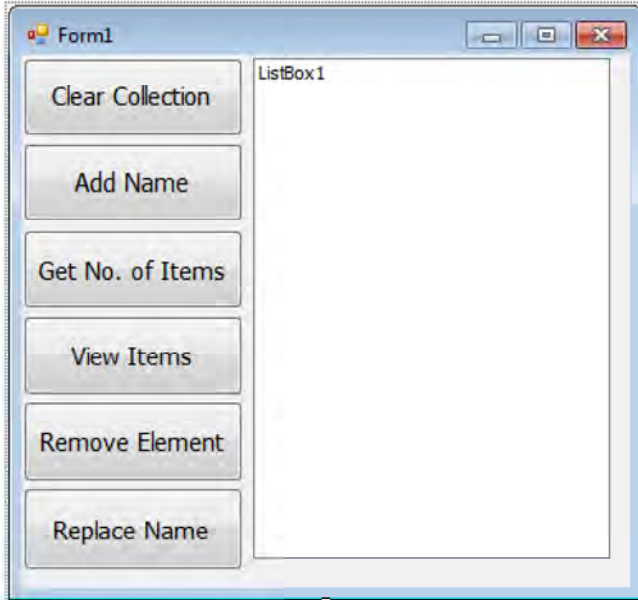
এখন আমরা একটি সাধারণ প্রোগ্রাম করবো। এতে কালেকশনের ব্যবহার সুস্পষ্ট হবে।

৪র্থ পরিচ্ছেদ: ভিজুয়াল বেসিক.নেট ভাষার **Variable Declaration**

ভিজুয়াল বেসিক.নেট : **Variable & Data Type**

CollectionDemo Application

১. প্রথমে CollectionDemo নামকরণে একটি ফর্ম প্রজেক্ট তৈরি করে নিচের চিত্রের মতো ডিজাইন করুন। এতে থাকবে ৬টি বোতাম ও ১টি লিস্ট বাক্স। দেখুন নিচের চিত্রটি (চিত্র: ২০)।



লক্ষ করুন, প্রতি বোতামের নামকরণ হবে দেখানো নামগুলোর মতো। লিস্ট বাক্সের নাম যা আছে তাই থাকবে। এবার নিচের কোডগুলো বর্ণিত বোতামের ক্লিক ইভেন্টে লিখুন।

১. Collection Declaration: এটা হবে ফর্মের ক্লাস ডিক্লারেশনের পরে:

```
Public Class Form1
    Dim MyCollection As New Collection
End Class
```

চিত্র -২০: CollectionDemo প্রজেক্টের ফর্ম

২. যথাক্রমে 'Clear Collection', থেকে 'Replace Name' বোতামের ক্লিক ইভেন্ট কোডসমূহ:

```
Public Class Form1
```

```
    Dim MyCollection As New Collection
```

```
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
```

```
        ' this method clears all the elements in the collection
```

```
        MyCollection.Clear()
```

```
    End Sub
```

```
    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
```

```
        ' read a name
```

```
        Dim Name As String
```

```
        Name = InputBox("Enter a Name") 'ask user to enter a name
```

```
        ' add the name into the list
```

```
        MyCollection.Add(Name)
```

```
    End Sub
```

```
    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
```

```
        'Show number of items in a message box.
```

```
        MsgBox("The number of items is: " & MyCollection.Count)
```

```
    End Sub
```

৪র্থ পরিচ্ছেদ: ভিজুয়াল বেসিক.নেট ভাষার **Variable Declaration**

ভিজুয়াল বেসিক.নেট : **Variable & Data Type**

CollectionDemo Application

Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click

' clear old content

ListBox1.Items.Clear()

' insert the items into the list box

Dim I As Integer

For I = 1 To MyCollection.Count

 ListBox1.Items.Add(MyCollection(I))

Next

End Sub

Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click

' get element position

Dim I As Integer

I = InputBox("Enter the element number you want to remove:") 'ask user to enter element number

If I <= MyCollection.Count And I >= 1 Then 'check in the element number is valid

 MyCollection.Remove(I)

Else

 'if invalid number entered then show a message, reminding of number of elements

 MsgBox("You have 1 to " & MyCollection.Count & " items in the collection.")

End If

End Sub

Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click

' get element position

Dim I As Integer

Dim N As String

I = InputBox("Enter the element number:") 'ask user to enter element number

If I <= MyCollection.Count And I >= 1 Then 'check in the element number is valid

 N = InputBox("Enter the new name:") 'ask user to enter new name

 MyCollection.Add(N, , I) 'add name in the element number

 MyCollection.Remove(I) 'remove old element

Else

 'if invalid number entered then show a message, reminding of number of elements

 MsgBox("You have 1 to " & MyCollection.Count & " items in the collection.")

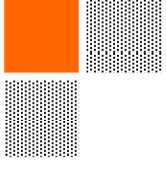
End If

End Sub

End Class

উপরের কোড দ্বারা আমরা কালেকশন ডিক্লয়ার করা, ক্লিয়ার করা, নতুন ইলিমেন্ট যোগ দেওয়া, ক'টি ইলিমেন্ট আছে তা জানা, সবগুলো প্রদর্শন করা, মুছে দেওয়া এবং পরিবর্তন করার পদ্ধতি দেখিয়েছি। পরের পৃষ্ঠায় প্রোগ্রাম রানিং অবস্থার একটি চিত্র (চিত্র : ২১) দিয়েছি। একটু দেখে নিন।

৫ম পরিচ্ছেদ: *Looping in Program*



বিভিন্ন লুপিং : *Various Looping*

আমরা ইতোমধ্যে ভিজুয়াল বেসিক.নেট এর কোডিংয়ে For Next লুপিং এর ব্যবহার দেখেছি। এটার আরেকটি উদাহরণ হলো:

Dim I As Integer

For I = 1 To 20

ListBox1.Items.Add(I)

Next

এই ফর-নেক্সট লুপ ১ থেকে ২০ পর্যন্ত সংখ্যা লিস্ট বাক্স ১-এ প্রদর্শন করবে। প্রত্যেকবার লুপিংয়ের সময় I-এর মাত্রা ১ বাড়তে থাকে। মার্কের লাইন এই আই I-এর মাত্রা লিস্ট বাক্সে লিখে দেয়। I-এর মান ২০ হওয়ার পর আরেকবার লুপিংয়ের জন্য ফিরে আসলে I-এর মান হবে ২১। সুতরাং লুপিং শেষ হবে, কারণ নির্দেশ দেওয়া আছে I হবে ১ থেকে ২০ -এর বেশিও না কমও না। এবার আমরা আরেক ধরনের লুপিং নিয়ে আলোচনা করবো।

Do Loop

এই লুপের নিয়ম নিম্নরূপ:

Do

statement

statement....

Loop While condition

অথবা

Do

statement.....

statement....

Loop Until condition

অথবা

Do While condition

statement...

statement...

Loop

অথবা

Do Until Condition

statement...

statement...

Loop

লক্ষ করুন: কন্ডিশন কোথায় আছে তা-ই নির্ধারণ করে কখন চেক করা বা বের হওয়া কিংবা থাকা নির্দিষ্ট বৈধ হবে। যেমন:

৫ম পরিচ্ছেদ: *Looping in Program*

বিভিন্ন লুপিং : *Various Looping*

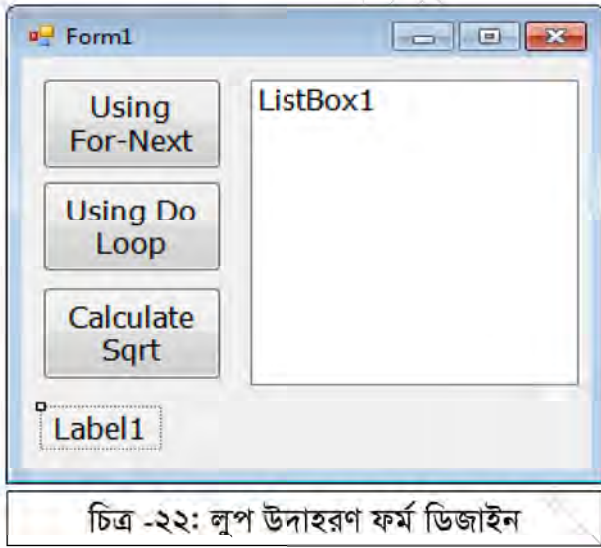
Do While L<10
statement ...
statement ...
Loop

এখানে প্রত্যেকবার চেক হবে লুপের কাজ শুরুর পূর্বে। এই লুপ কখনো শুরু না হওয়ার সম্ভাবনা আছে।

Do
statement ...
statement ...
Loop While L<10

এখানে চেক হবে লুপের কাজ শুরুর পরে। এই লুপ অন্তত একবার শুরু হবে।

এবার একটি উদাহরণ দেখে নিই। নিচের ফর্মের মতো একটি ফর্ম ডিজাইন করুন। এতে থাকবে তিনটি বোতাম, একটি লিস্ট বাক্স ও একটি লেবেল।



চিত্র -২২: লুপ উদাহরণ ফর্ম ডিজাইন

প্রথম বোতামের কোড হবে:

```
Dim I As Integer 'declare an integer variable
ListBox1.Items.Clear() 'empty out list box
For I = 1 To 10
    ListBox1.Items.Add(I) '1 to 10 into list box
Next
```

দ্বিতীয় বোতামের কোড হবে:

```
Dim I As Integer 'declare integer variable
I = 1 'initialize it
ListBox1.Items.Clear() 'clear items in the list box
Do
    ListBox1.Items.Add(I) 'add 1 to 10 in the list box
    I = I + 1 'add 1 to I
Loop While (I < 11) 'check. I must be 11 to exit the loop
```

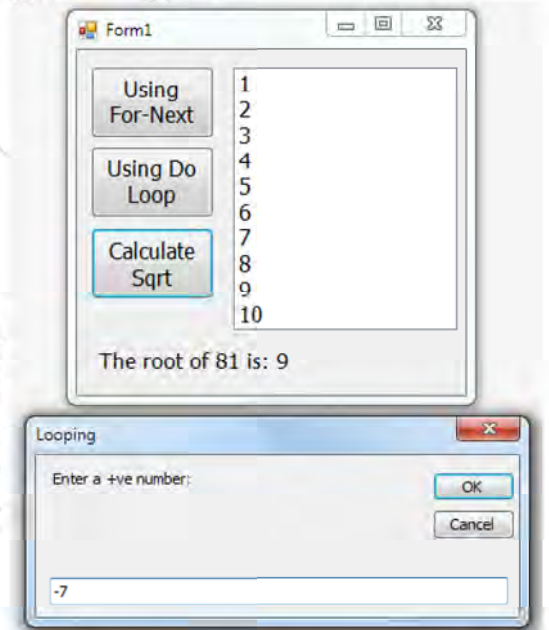
তৃতীয় বোতামের কোড হবে:

```
Dim I As Double 'declare double type variable (decimal number).
Do
I = InputBox("Enter a +ve number:") 'ask for a positive number.
'calculate sqrt and display in the label
If I >= 0 Then Label1.Text = "The root of " & I & " is: " &
Math.Sqrt(I)
Loop
```

লক্ষ করুন: প্রথম দু'টি বোতামের লুপ কোড এমনতেই শেষ হবে। কিন্তু তৃতীয় বোতামের লুপ কোড ব্যবহারকারী নেগেটিভ সংখ্যা দ্বারা শেষ করবেন। অন্যথায় এই লুপ চিরকাল চলতে থাকবে! একে বলে নেভার এন্ডিং লুপ [Never ending loop]। প্রোগ্রামারকে এ ব্যাপারে সতর্ক থাকা চাই। অনাকাঙ্ক্ষিত নেভার এন্ডিং লুপে চলে গেলে কম্পিউটার ক্রাশ করতে পারে। যেমন আমরা যদি লিখে ফেলি (এটা করবেন না!):

```
dim a as integer =1
For b = 1 to a
    a = a+1
Next b
```

এখানে মাঝের লাইন দ্বারা a-এর মাত্রা এক এক করে বাড়তে থাকবে। সুতরাং b-এর মাত্রা কখনও শেষ হবে না এবং লুপও ক্লোজ হবে না। কম্পিউটার ক্রাশ করবে। উপরের লুপে `If I >= 0 Then` না থাকলে প্রোগ্রাম বন্ধ করা যেতো না।



চিত্র -২৩: লুপ উদাহরণ রানটাইম

৬ষ্ঠ পরিচ্ছেদ: *Structure*



বিভিন্ন ভেরিয়েবল একত্রে : *Different Variables together*

একাধিক ভিন্ন ভেরিয়েবল নিয়ে একেকটি Structure তৈরি হয়। এটা মূলত প্রোগ্রামার দ্বারা বিভিন্ন ভেরিয়েবল একটি ইউনিট হিসাবে সৃষ্টি করার সুযোগ যা অনেকটা সুবিধাসম্পন্ন। উদাহরণ দ্বারা বিষয়টি পরিষ্কার হবে।

মনে করুন আপনি কোনো কোম্পানির কর্মচারীদের বিভিন্ন তথ্য একত্রে সংরক্ষণ করতে চান। তথ্যাদি হবে নাম, ঠিকানা, টেলিফোন নাম্বার, বয়স, বেতন, কাজের নাম ইত্যাদি। ইতোমধ্যে আমরা যা জেনেছি তার দ্বারা এটাই নির্দেশ করে যে প্রতিটি ভিন্ন তথ্য একেকটি আলাদা ভেরিয়েবলে সংরক্ষণ করতে হবে। যেমন:

Dim EName as String, Dim EAddress as String, Dim ETelephone as String, Dim EAge as Integer, Dim ESalary as Decimal, Dim EJobTitle as String

এখন অপর আরেক কর্মচারীর জন্য অনুরূপভাবে ভেরিয়েবল ডিক্লারেশন দরকার। অন্য আরেক জনের জন্যও এরূপ করতে হবে। সুতরাং দশজন কর্মচারী হলে ৬০ টি ভেরিয়েবল লাগবে। বেশি হলে আরো বেশি লাগবে। এভাবে প্রোগ্রামিং করলে আপনি শীঘ্রই ইস্তিফা দেবেন! আসলে এরূপ করার দরকার নেই। আপনি Structure ব্যবহার করে একবার মাত্র ভেরিয়েবল ডিক্লারেশন করে কাজ সারতে পারেন। কিভাবে? এই দেখুন:

Structure ডিক্লারেশন

বিভিন্ন প্রোপার্টি একত্রে একটি Structure-এ এভাবে ডিক্লারেশন করা যায়:

```
Structure EmployeeInfo
    Dim EName as String
    Dim EAddress as String
    Dim ETelephone as String
    Dim EAge as Integer
    Dim ESalary as Decimal 'decimal number data type
    Dim EJobTitle as String
End Structure
```

এখন আমরা নতুন একটি ডাটা টাইপ তৈরি করলাম যার নাম EmployeeInfo। এর ভেতর আছে ৬টি ভিন্ন তথ্য। এবার EmployeeInfo-কে আমরা ডাটা টাইপ হিসাবে ডিক্লারার করতে পারি যেমনটি পারি String, Integer ইত্যাদি। সুতরাং আমরা ডিক্লারার করবো এভাবে:

```
Dim A as EmployeeInfo
A.EName = "Muhammad Abdus Samad"
A.EAddress = "10 Bashundara R/A, Maulvi Bazar, Bangladesh"
A.ETelephone = "01235 215226"
A.EAge = 35
A.ESalary = 32500.00
A.EJobTitle = "Managing Director"
```

স্টোর করা তথ্য আনতে হবে এভাবে: A.ETelephone, A.ESalary = 32500.00 ইত্যাদি। এ হলো কর্মচারী A। আমরা একই ভেরিয়েবল টাইপ ব্যবহার করে আরো কর্মচারীর তথ্য সংরক্ষণ করতে পারি। যেমন:

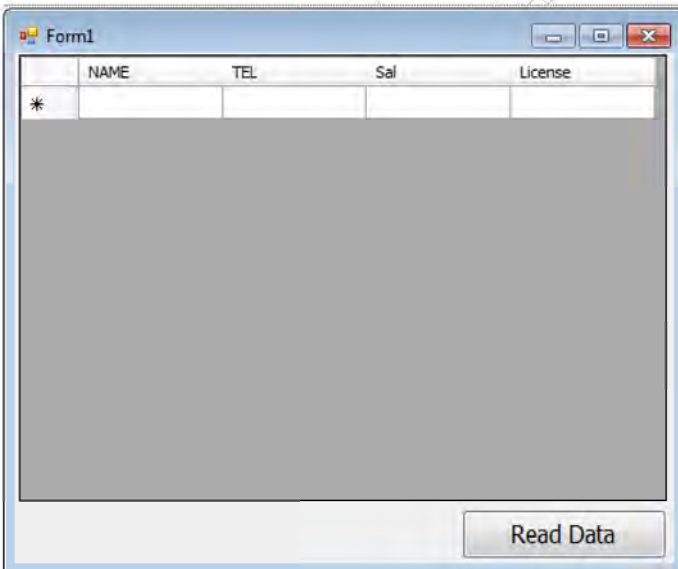
৬ষ্ঠ পরিচ্ছেদ: *Structure*

বিভিন্ন ভেরিয়েবল একত্রে : *Different Variables together*

```
Dim B as EmployeeInfo
B.ENAME = "Abdur Rahim"
B.EAddress = "99 Sunamgonj Rd, Patantula R/A, Sylhet, Bangladesh"
B.ETelephone = "01723658211"
B.EAge = 33
B.ESalary = 22500.00
B.EJobTitle = "Chief Supervisor"
```

অনুরূপ আমরা যতোজন কর্মচারীর প্রয়োজন ততোজনের তথ্য EmployeeInfo এর মধ্যে সংরক্ষণ করতে পারি। সুতরাং বুঝাই যাচ্ছে Structure একটি গুরুত্বপূর্ণ ডিক্লারেশন। এবার আমরা Structure ব্যবহারের একটি উদাহরণ দেখাবো। নিচের পদক্ষেপগুলো অনুসরণ করুন।

১. নতুন প্রজেক্ট তৈরি করুন। ফর্মের মধ্যে যোগ করুন ১টি বোতাম ও একটি ডাটাগ্রিডভিউ [DataGridView]। গ্রিডে থাকবে: Name, Tel, Sal, License -এ চারটি কলাম। দেখুন নিচের চিত্রটি।



চিত্র -২৪: স্ট্রাকচার প্রজেক্টের ফর্ম

২. ফর্মের যে কোনো স্থানে ডবল ক্লিক করুন। কোড জানালা আসবে। এতে লিখুন নিচের কোডটি:

```
Public Class Form1
    'declare the structure
    Structure PersonInfo
        'person name
        Dim Name As String 'person telephone number
        Dim Tel As String
        'person salary
        Dim Sal As Decimal
        'person license number
        Dim LincenseNumber As String
    End Structure
    'this is the array to store persons' info
    Dim Info() As PersonInfo
End Class
```

এ কোড ব্লক দ্বারা আমরা PersonInfo নামে একটি নতুন ভেরিয়েবল ডিক্লার করার করেছি। এ ব্লকটি ফর্মের ক্লাস ডিক্লারেশন পরেই থাকবে। আলাদা কোনো সাবরুটিনের ভেতর একে ডিফাইন করা যাবে না। PersonInfo হলো একটি Structure ভেরিয়েবল। এ ভেরিয়েবলের ভেতর কয়েকটি ভিন্ন ভেরিয়েবল ডিক্লারেশন আছে। এ ক্ষেত্রে এগুলো হলো: Name, Tel, Sal এবং LicenseNumber। লক্ষ করুন, এসব ভেরিয়েবলের তিনটি String Type ও একটি Decimal Type ভেরিয়েবল ডিক্লারেশন করা হয়েছে। স্ট্রাকচারের মধ্যে এভাবে ভেরিয়েবল ডিক্লারেশনের ফলে এটা সম্ভব হয়েছে। প্রোগ্রামের সর্বত্র পাওয়ার নিশ্চয়তার জন্য Info() নামক একটি PersonInfo টাইপ এ্যারে ভেরিয়েবলও এখানে ডিক্লার করার করেছি। এই এ্যারের মধ্যে ব্যক্তির তথ্যাদি সংরক্ষণ করা হবে। এখন আমাদেরকে PersonInfo নামক ভেরিয়েবলে তথ্য ঢুকাতে হবে। এজন্য আমরা Read Data বোতামটি ব্যবহার করবো। এর ক্লিক ইভেন্ট হ্যান্ডলারে আমরা পরের পৃষ্ঠার কোডটি লিখবো:

৬ষ্ঠ পরিচ্ছেদ: *Structure*

বিভিন্ন ভেরিয়েবল একত্রে : *Different Variables together*

```
' read the information and store it in an array
Public Sub ReadInfo()
    Dim I As Integer 'I for For-Next Loop
    Dim N As Integer 'N for number of persons
    On Error GoTo ChkNum ' incase the user enters a wrong entry [not number]
    N = InputBox("Enter the number of people") 'first enter how many person
    ReDim Info(0 To N - 1) 'redimension the array to 0 to number of person
    ' read info here
    For I = 0 To N - 1
        Info(I).Name = InputBox("Enter the name of person")
        Info(I).Tel = InputBox("Enter the telephone number")
        Info(I).Sal = InputBox("Enter the salary")
        Info(I).LincenseNumber = InputBox("Enter license number")
    Next
Exit Sub
ChkNum: 'label jump
Exit Sub
End Sub
```

উক্ত সাবটি কিভাবে কাজ করে তা আপনার মোটামুটি জানা হয়ে গেছে আশা রাখি। এরপরও আবার বুঝিয়ে দিচ্ছি। প্রথমত সাবরুটিনটি নিজের বানানো মনে রাখবেন। আপনাকে সবকিছু নিজেই টাইপ করতে হবে। প্রথম লাইন দ্বারা ইন্টেজার ভেরিয়েবল ডিক্লারেশন করা হয়েছে। এই ইন্টেজার ফর-নেস্ট লুপের গণক হিসাবে ব্যবহৃত হয়েছে। দ্বিতীয় লাইনে ঘোষিত ইন্টেজার হলো লোকসংখ্যা। এটার মাত্রা চতুর্থ লাইনে নিশ্চিত হবে। ব্যবহারকারী যতো চুবাকেন তা-ই হবে এর মাত্রা। তৃতীয় লাইনটিকে বলে Error Handling। ভুলবশত ব্যবহারকারী সংখ্যা না চুকিয়ে অন্য কিছু চুকালে এই লাইন নিশ্চিত করবে সব থেকে বের হতে। অন্যথায় প্রোগ্রাম ক্রাশ করবে। এই লাইন লেভেলে জাম্প করবে।

এবার সংগ্রহিত তথ্যাদি ডাটা-গ্রিড ভিউ-এ প্রদর্শনের প্রয়োজন। এজন্য আমরা নিজে একটি সাবরুটিন বানিয়েছি। এটি ফাংশনের মতো কাজ করবে। যখনই আমরা একে কল করবো তখনই এটা সক্রিয় হবে।

```
' fill the data grid with array info
Public Sub FillDGV(ByVal DAT() As PersonInfo, ByVal DGV As DataGridView)
    DGV.Rows.Clear()
    Dim I As Integer
    For I = 0 To DAT.Length - 1
        On Error GoTo endSub 'error handling. Jump to endSub label and exit
        DGV.Rows.Add(DAT(I).Name, DAT(I).Tel, DAT(I).Sal, DAT(I).LincenseNumber)
    Next
Exit Sub
endSub: 'label jump
MsgBox("Enter a whole number please!")
Exit Sub
End Sub
```

উপরের সাব কিভাবে কাজ করে তা একটু তলিয়ে দেখা যাক। প্রথমত জেনে রাখি, এই সাবটি কল করা হবে নিচের দু'টি বোতামের সাব থেকে। কল করলেই এটা কাজ শুরু করবে। প্রথম লাইন তথা DGV.Rows.Clear() সকল রো ক্লিয়ার করবে। দ্বিতীয় লাইনে আমরা ফর-নেস্ট লুপের জন্য একটি কন্ট্রোল ইন্টেজার ডিক্লারেশন করেছি। লক্ষ করুন: ক'বার ফর-নেস্ট লুপ ঘুরবে তা নির্ভর করে ঠিক ক'জন ব্যক্তির তথ্য আমাদের এ্যারের মধ্যে সংরক্ষিত আছে- তার ওপর। লুপের লাইনটিতে DAT(I).Name, DAT(I).Tel, DAT(I).Sal, DAT(I).LincenseNumber দ্বারা একে একে সবগুলো রো পূর্ণ হবে। এবার দেখে নিই বাকি কোডগুলো।

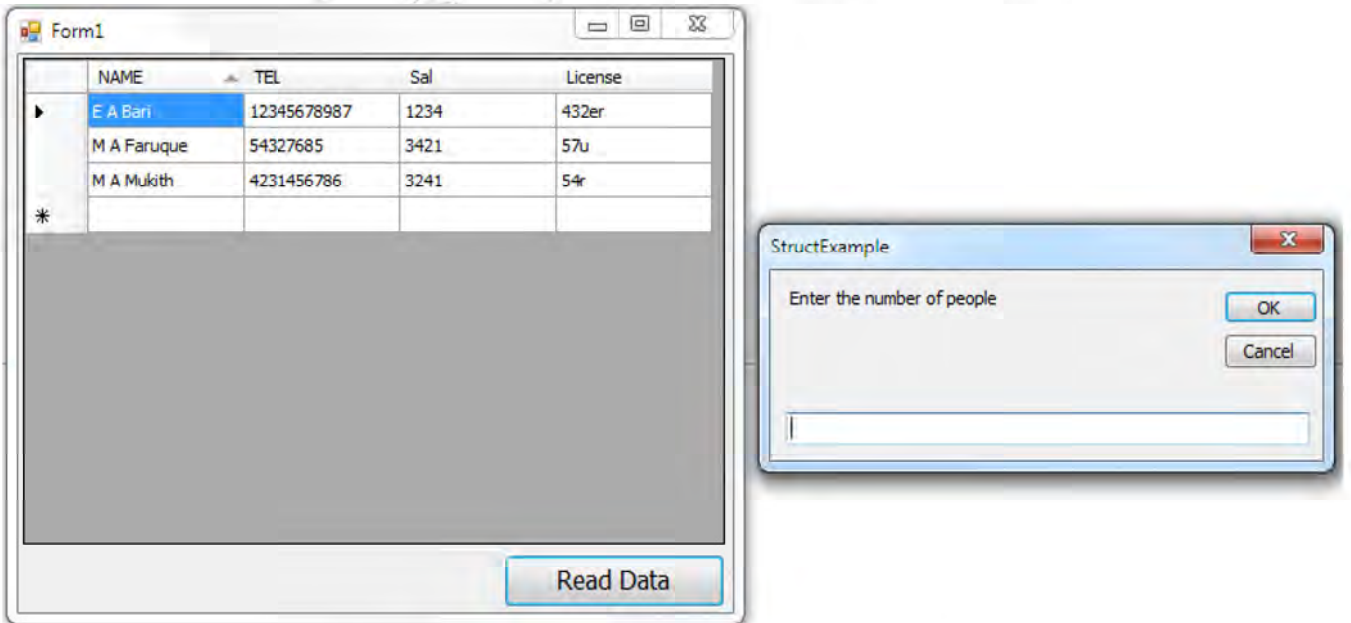
৬ষ্ঠ পরিচ্ছেদ: *Structure*

বিভিন্ন ভেরিয়েবল একত্রে : *Different Variables together*

Read Data বোতাম দ্বারা আমরা ব্যক্তিদের তথ্য সংগ্রহ করবো। এই বোতামের ক্লিক ইভেন্টে নিচের কোডটি লিখতে হবে। এই বোতাম দ্বারা ব্যবহারকারী প্রয়োজন পরিমাণ তথ্য ঢুকানোর পর গ্রিডে সবকিছু প্রদর্শিত হবে।

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    ReadInfo() 'call sub to enter data by user
    FillIDGV(Info, DataGridView1) 'call sub to display data in the grid
End Sub
```

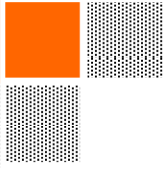
লক্ষ করণ এই সাবটির মধ্যে মাত্র দু'টি লাইন আছে। উভয় লাইন দ্বারাই আমরা ইতোমধ্যে ব্যাখ্যাত নিজের তৈরি সাব কল করেছি। প্রথমে আমার কল করেছি ReadInfo() সবার। এটা ব্যবহারকারীকে তথ্য ঢুকানোর জন্য ডায়ালগ প্রদর্শন করবে। তথ্য ঢুকানো শেষ হলেই দ্বিতীয় সাব FillIDGV(Info, DataGridView1) কল করা হবে। এ সাবের কাজ হলো তথ্যাদি গ্রিডে প্রদর্শন করা। ব্যস! স্ট্রাকচার ব্যবহারের উদাহরণ প্রোগ্রাম শেষ হলো। এবার চালু করে দেখার পালা। নিচের চিত্রটি দেখুন। লক্ষ করুন, NAME-এ ক্লিক করলে সমগ্র তথ্য আক্ষরিকভাবে সোর্ট হবে।



চিত্র -২৫: স্ট্রাকচার প্রোগ্রাম চলাকালীন চিত্র

পরের পরিচ্ছেদে আমরা ভিজুয়াল বেসিক.নেট এর আরো একটি গুরুত্বপূর্ণ বিষয়ের উপর উদাহরণসহ আলোচনা করবো। এ বিষয়টি হলো অন্যত্র কোড লিখে সংরক্ষণ ব্যবস্থা। এ পর্যন্ত আমরা যা করেছি তা সবই ফর্মের ক্লাস সাবরুটিনের ভেতর ছিলো। এখন একটি কোড ব্লকের উপর আলোচনা করবো যা লিখা হবে বাইরে- আলাদাভাবে। তবে অবশ্যই এটা যে কোনো জায়গায় ব্যবহারযোগ্য হবে। এই ব্যবস্থার নাম মডিউল [Module]।

৭ম পরিচ্ছেদ: *Separate code block in Program*



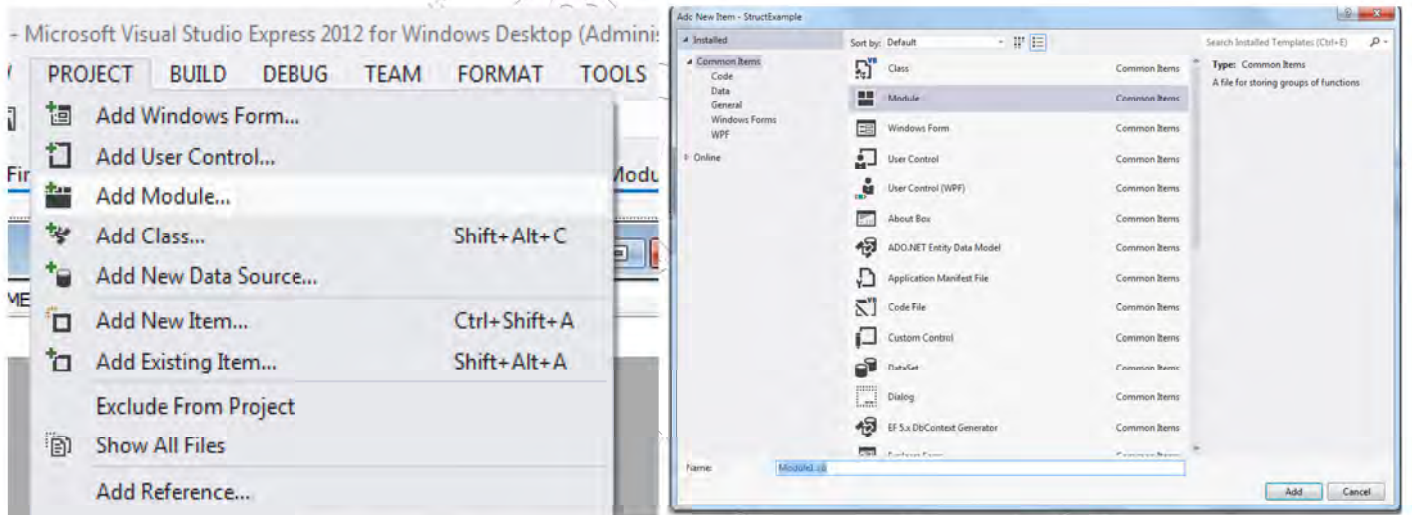
মডিউল ও ক্লাস : *Module & Class*

ভিজুয়াল বেসিক.নেটে আপনি আলাদাভাবে কোড ফাইল লিখে যে কোনো স্থানে সংরক্ষণ করতে পারেন। এটা খুব সুবিধাজনকও বটে। আপনি একই কোড বিভিন্ন প্রোগ্রামে ব্যবহার করতে চাইলে এ সুবিধা থাকাটা খুব জরুরীও।

মডিউল [Module]

মডিউলে শুধুমাত্র কোড থাকে। অর্থাৎ ফাংশন, সাব, স্ট্রাকচার ইত্যাদি ছাড়া আর কিছুই থাকবে না। এতে না আছে কোনো উইন্ডো, ফর্ম, বোতাম কিংবা গ্রাফিক্যাল ইউজার ইন্টারফেইস [GUI]। মডিউলে থাকবে শুধুমাত্র কোড।

প্রজেক্টে মডিউল যুক্ত করতে হলে প্রজেক্ট [Project] মেন্যুতে যেয়ে এড মডিউল [Add Module] চয়েজ করতে হবে। দেখুন নিচের ছবিটি (বায়ে)। Add Module চয়েজ করলে আসবে আরেকটি জানালা। ডানের চিত্রটি দেখুন।



এভাবে নতুন মডিউল বানিয়ে প্রজেক্টে যুক্ত করার পর মডিউলের কোড জানালা আসবে। আপনি এতে সাব, ফাংশন, স্ট্রাকচার, ইত্যাদি ডিক্লার করার করতে পারেন। এসব বস্তু প্রোগ্রামের যে কোনো স্থান থেকে কল করে ব্যবহার করতে পারেন। সর্বোপরি এই মডিউল অন্য কোনো প্রজেক্টেও নিতে পারেন। আর মডিউল শব্দের অর্থই এটা। নিচে মডিউলের উদাহরণ কোড দেখিয়েছি। এর সাব ও ফাংশন কিংবা যে কোনো একটি প্রোগ্রামের অন্যত্র থেকে কল করতে পারেন। একটু স্টাডি করে নিন।

Public Module ThisModule

Sub Main()

Dim userName As String = InputBox("What is your name?") 'ask for user name

MsgBox("User name is: " & userName) 'show it in message box

End Sub

Function Sq(SupNum As Double) As Double 'returns square of number

Return SupNum * SupNum

End Function

' Insert variable, property, procedure, and event declarations.

End Module

মডিউল ও ক্লাসের ব্যবহার এ পরিচ্ছেদের শেষে একটি উদাহরণে দেখিয়েছি।

৭ম পরিচ্ছেদ: *Separate code block in Program*

মডিউল ও ক্লাস : *Module & Class*

'this three subs makes sure only decimal numbers are entered into textBox1, TextBox2 & TextBox3

```
Private Sub TextBox1_KeyPress(sender As Object, e As KeyPressEventArgs) Handles TextBox1.KeyPress
e.Handled = Not (Char.IsDigit(e.KeyChar) Or Asc(e.KeyChar) = 8 Or ((e.KeyChar = ".") And (sender.Text.IndexOf(".") = -1)))
End Sub
Private Sub TextBox2_KeyPress(sender As Object, e As KeyPressEventArgs) Handles TextBox2.KeyPress
e.Handled = Not (Char.IsDigit(e.KeyChar) Or Asc(e.KeyChar) = 8 Or ((e.KeyChar = ".") And (sender.Text.IndexOf(".") = -1)))
End Sub
Private Sub TextBox3_KeyPress(sender As Object, e As KeyPressEventArgs) Handles TextBox3.KeyPress
e.Handled = Not (Char.IsDigit(e.KeyChar) Or Asc(e.KeyChar) = 8 Or ((e.KeyChar = ".") And (sender.Text.IndexOf(".") = -1)))
End Sub
```

এবার প্রোগ্রামটি রান করে দেখুন। প্রথম তিনটি বাক্সে সংখ্যা ছাড়া অন্য কিছু লিখার চেষ্টা করুন। পারবেন না। তবে সংখ্যা [দশমিক বিন্দুসহ] ঠিকই লিখতে পারবেন। উপরের কোডগুলো গুরুত্বপূর্ণ। সংখ্যা ছাড়া অন্য কিছু থাকলে প্রোগ্রাম ত্রাশ করবে- যা আমাদের ইচ্ছে নেই।

৬. প্রোগ্রাম বন্ধ করে প্রজেক্ট মেন্যু থেকে চয়েজ করুন Add Class। ক্লাসের নামকরণ করুন: ClassBox.vb।

৭. ক্লাসের কোড হবে নিম্নরূপ:

```
Public Class ClassBox
    Public length As Double ' Length of a box
    Public width As Double ' Width of a box
    Public height As Double ' Height of a box
End Class
```

৭. এবার প্রজেক্ট মেন্যু থেকে চয়েজ করুন Add Module। মডিউলের নামকরণ করুন: BoxMod.vb। কোড হবে:

```
Module BoxMod
    'define Volume Function here / public function available anywhere in the programme
    Public Function Vol(SupVL As Double, SupVW As Double, SupVH As Double) As Double
        Return SupVL * SupVW * SupVH
    End Function
End Module
```

৮. ডবল ক্লিক করুন Calculate Box Volume বোতামে। প্রদর্শিত সাবে লিখুন:

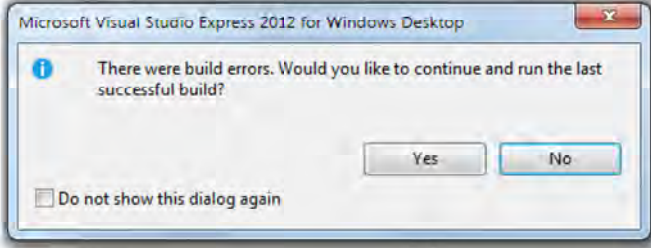
```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    'make sure user entered numbers
    If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Then
        Exit Sub
    End If
    Dim Cl As New ClassBox 'declare a class type
    Cl.length = TextBox1.Text 'get value
    Cl.width = TextBox2.Text 'get value
    Cl.height = TextBox3.Text 'get value
    'Make sure they are numbers - in case user enters just space. Although in our example space not possible
    If IsNumeric(Cl.length) = False Or IsNumeric(Cl.width) = False Or IsNumeric(Cl.height) = False Then
        Exit Sub
    End If
    Dim ResVal As Double = Vol(Cl.length, Cl.width, Cl.height)
    TextBox4.Text = ResVal
End Sub
```

৭ম পরিচ্ছেদ: *Separate code block in Program*

মডিউল ও ক্লাস : *Module & Class*

Compile and Test

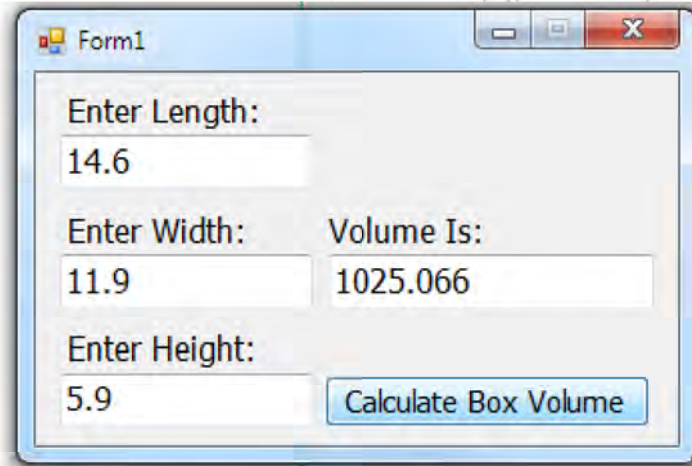
ইতোমধ্যে আমরা প্রজেক্ট Start করে বার বার প্রোগ্রাম চালিয়ে পরীক্ষা করেছি। বাস্তবে Start মেন্যুটি চয়েজ করার পর কী হয়? সকল কোড প্রথমে এক্সপ্রেস ২০১২ বাইনারি কোডে পরিণত করে- একে বলে Compile। বাইনারি কোড মানে ০ এবং ১। আর ০ মানে সুইচ অফ এবং ১ মানে সুইচ অন। কম্পাইলার প্রথমে কোডের প্রতিটি লাইন একে একে পরীক্ষা করে দেখে এতে কোনো ভুল বা সিনটেক্সট এরোর আছে কি না। যদি থাকে তাহলে কম্পাইলের পূর্বে একটি ওয়ার্নিং ডায়ালগ আসবে। ডায়ালগটি নিম্নরূপ:



সাধারণত আপনি 'No' বোতামে ক্লিক করবেন। তখন ভুলটি কোথায় হয়েছে তা-ও দেখাবে। ঠিক করার পর আবার Start Menu চয়েজ করে কম্পাইল করবেন। আপনার প্রোগ্রাম সম্পূর্ণ হয়ে গেলে কম্পাইল করে একটি .exe ফাইল তৈরি করবেন। এই কম্পাইল করা ফাইলই হলো এ্যাপলিকেশনের সর্বশেষ রূপ। একে

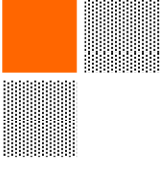
পরিবর্তন করার কোনো সুযোগ আর থাকবে না। অন্যকথায় ডিস্ট্রিবিউশন পরে ব্যবহারকারী এই ফাইলে ডবল ক্লিক করে প্রোগ্রাম চালাবে। ব্যস! ভুল ত্রুটি সংশোধন একমাত্র আপনার কাজ। একে বলে ডিবাগিং। আপনি পরবর্তী ভার্শনে সংশোধিত আরেকটি .exe ফাইল [প্রয়োজনে Installer-সহ] পুনঃরিলিজ করবেন।

আপনার প্রোগ্রাম যদি বেশি জটিল হয় তাহলে বিল্ড মেন্যু [BUILD] চয়েজ করে আপনি 'Publish' [আপনার প্রোগ্রামের নাম] চয়েজ করবেন। এরপর প্রদর্শিত নির্দেশনাবলী মূতাবিক পাবলিশ করে সংরক্ষণ করবেন। যাক, এ ব্যাপারে গ্রন্থের শেষের দিকে আরো বিস্তারিত আলোচনা হবে। এখন আমরা এইমাত্র সম্পন্ন ModClassExample প্রোগ্রামটি Start করে টেস্ট করবো। দেখুন নিচের চিত্রটি।



পরের পরিচ্ছেদে আমরা Sub এবং Function নিয়ে উদাহরণসহ অতিরিক্ত কিছু আলোচনার প্রয়াস পাচ্ছি। এর কারণ হলো, সকল ডিজিটাল বেসিক.নেট মূলত এ দু'টোকে নিয়েই তৈরি হয়। সুতরাং এগুলো সম্পর্কে যা-ই আমরা জানবো তা-ই লাভজনক হবে।

৮ম পরিচ্ছেদ: *More on Subs & Functions*



সাব ও ফাংশন : *Subs & Functions*

আমরা ইতোমধ্যে যাকিছু কোডিং করেছি তা সবই ছিলো মূলত সাব ও ফাংশন। হ্যাঁ, আমরা মডিউল এবং ক্লাস সম্পর্কেও জেনেছি। কিন্তু এগুলো আলাদা কোড ব্লক ছাড়া আর কিছু নয়। এগুলোর ভেতরও থাকে সাব এবং ফাংশন। সুতরাং সাব ও ফাংশন সম্পর্কে বেশি জানার অর্থ হলো .নেট কোডিং সম্পর্কে আরো বেশি অবগত হওয়া।

Writing Sub Procedure -সাব প্রসিডিউর লেখা

টেকনিক্যালী সাবরুটিনকে সাব প্রসিডিউর বলে। প্রত্যেক সাব প্রসিডিউর কল করা হয় তার নাম দিয়ে। এ কল হতে পারে অন্য কোনো ইভেন্ট সাব প্রসিডিউর থেকে কিংবা অন্য কোনো ইউজার-ডিফাইন্ড [অর্থাৎ আপনার বানানো] সাব প্রসিডিউর থেকে। এরা এক বা একাধিক তথ্য অথবা আর্জুমেন্ট গ্রহণ করে এর ওপর কাজ করতে পারে। কাজ থেকে ফলাফল ফেরৎ পাঠাতেও পারে। কিন্তু ফাংশনের মতো নামের সাথে ভ্যালু বা মান রিটার্ন করতে পারে না। অন্যকথায় সাবরুটিন ব্যবহৃত হয়: ১. ইনপুট (প্রদত্ত ভ্যালু) গ্রহণ বা প্রসেস করতে, ২. আউটপুট (ফলাফল) প্রদর্শন করতে ও ৩. প্রোপার্টিজ [বৈশিষ্ট্যাবলী] সেট করতে।

এবার একটি উদাহরণ দেখবো। সাব প্রসিডিউর প্রায়ই ব্যবহৃত হয় ইনপুট নিয়ন্ত্রণে। এই উদাহরণে আমরা 'AddName' নামক একটি Sub Procedure তৈরি করবো। এটা ব্যবহারকারীকে লেখা প্রবেশের জন্য অনুরোধ জানাবে এবং তা টেক্সট বক্সে প্রদর্শন করে দেখাবে। প্রসিডিউরটি একটি মডিউলে থাকবে। ফলে একে একাধিকবার কল করা যাবে। মূলত পুরো প্রোগ্রামের যে কোনো স্থান থেকে একে কল করে ব্যবহার করা যেতে পারে।

১. একটি নতুন প্রজেক্ট তৈরি করুন। এর নামকরণ করুন: 'My Text Box Sub'।
২. ফর্মে দু'টি টেক্সট বক্স যুক্ত করুন।
৩. বাক্স দু'টোর উপরে দু'টি লেবেল যুক্ত করুন। এগুলোতে থাকবে ডিপার্টমেন্টের নাম।
৪. মোট তিনটি বোতাম যুক্ত করুন। প্রথম দু'টি যথাক্রমে টেক্সট বাক্স ১ ও টেক্সট বাক্স ২ এর নিচে রাখুন। বাকিটি রাখুর ফর্মের নিচের দিকে।
৫. নিচের টেবিল মুতাবিক বিভিন্ন বস্তুর প্রোপার্টি সেট করুন:

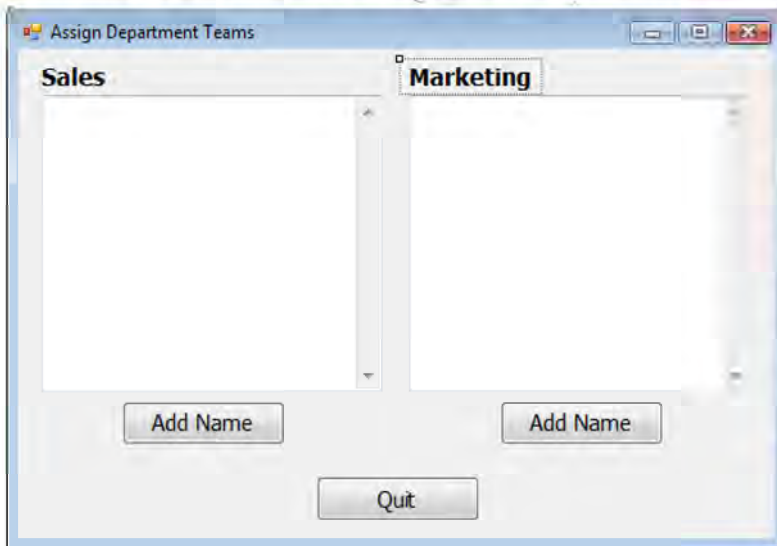
বস্তু	প্রোপার্টি	সেটিং
TextBox1	Multiline	True
	Name	txtSale
	ReadOnly	True
	ScrollBars	Vertical
	TabStop	False
	Font Size	12
TextBox2	All Same accept Name txtMkt	
Label1	Font	Bold
	Font Size	12
	Name	lblSales
	Text	"Sales"
Label1	Font	Bold
	Size Size	12
	Name	lblMkt
	Text	"Marketing"

৮ম পরিচ্ছেদ: *More on Subs & Functions*

সাব ও ফাংশন : *Subs & Functions*

বস্তু	প্রোপার্টি	সেটিং
Button1	Name	btnSales
	Text	"Add Name"
	Font Size	11
Button2	Name	btnMkt
	Text	"Add Name"
	Font Size	11
Button3	Name	btnQuit
	Text	"Quit"
	Font Size	11
Form1	Text	"Assign Department Teams"

সব ঠিকমতো হলে ফর্মটির ডিজাইন হবে নিচের চিত্রের মতো।



এখন আমরা প্রজেক্টে একটি মডিউল যুক্ত করবো। আর এতেই লিখবো আমাদের সর্বত্র ব্যবহারযোগ্য AddName সাবরুটিনটি।

৬. প্রজেক্ট মেন্যু থেকে চয়েজ করুন Add Module। নতুন মডিউল তৈরি হবে। এটার নামকরণ যা আছে তা-ই রেখে দিন।

৭. মডিউল তৈরির পরই এর কোড জানালা খুলবে। এই কোড জানালায় থাকবে নিচের কোড ব্লকটি:

```
Module Module1
    'Our Sub here
End Module
```

আমরা যে সাব প্রসিডিউর লিখবো তা এই ব্লকের ভেতরই থাকবে।

৭. নিচের কোডটি হবে মডিউলের মধ্যে।

Module Module1

```
Sub AddName(ByVal Team As String, ByRef ReturnString As String)
```

```
    Dim Prompt, Nm, WrapCharacter As String
```

```
    Prompt = "Enter a " & Team & " employee."
```

```
    Nm = InputBox(Prompt, "Input Box")
```

```
    WrapCharacter = Chr(13) + Chr(10)
```

```
    ReturnString = Nm & WrapCharacter
```

```
End Sub
```

```
End Module
```

ব্যাখ্যা: এই প্রসিডিউর কল হলে দু'টি আর্গুমেন্ট গ্রহণ করবে। এগুলো হলো Team, স্ট্রিং টাইপ ডিপার্টমেন্টের নাম এবং ReturnString, একটি শূন্য স্ট্রিং ভেরিয়েবল যাতে থাকবে কর্মকতার নাম। লক্ষ্য করুন, ReturnString-কে

৮ম পরিচ্ছেদ: *More on Subs & Functions*

সাব ও ফাংশন : *Subs & Functions*

ByRef কিওয়ার্ডসহ ডিক্লার করা হয়েছে। এর ফলে প্রসিডিউরে এই আর্গুমেন্টে কোনো পরিবর্তন হলে তা কলিং রুটিনে ফিরে দেওয়া হবে। প্রথম লাইনে আমরা তিনটি স্ট্রিং ভেরিয়েবল ডিক্লার করেছি। এগুলো ২য়, ৩য় ও ৪র্থ লাইনে ইনিশিয়েলাইজ করা হয়েছে। শেষের লাইনে রিটার্নস্ট্রিংকে ফর্মের ফর্মট করেছি। ব্যস! এটুকুই। এই প্রসিডিউর ব্যবহারকালে বুঝতে পারবেন এটা কিভাবে কাজ করে।

৮. পুনরায় ফর্ম জিডাইন প্রদর্শন করুন। ডবল ক্লিক করুন ফর্মের Add Name বোতামে। বোতামের ক্লিক ইভেন্ট প্রসিডিউর খুলবে। এতে যুক্ত করুন নিচের কোডটি।

calls AddName Procedure defined in Module1

Private Sub btnSales_Click(sender As Object, e As EventArgs) Handles btnSales.Click

```
Dim SalesPosition As String = ""
AddName("Sales", SalesPosition)
txtSale.Text = txtSale.Text & SalesPosition
```

End Sub

ব্যাখ্যা: AddName প্রসিডিউর কলে দু'টি আর্গুমেন্ট আছে: Sales এবং SalesPosition। শেষের লাইনের কাজ হলো রেফারেন্সের মাধ্যমে প্রেরিত আর্গুমেন্ট ব্যবহার করে txtSale টেক্সট বাক্সে লেখা যুক্ত করা। কনক্যাটেনেশন অপারেটর & এর কারণে নতুন নাম টেক্সটের শেষে যুক্ত হবে। যাক, প্রোগ্রাম চলাকালে বিষয়টি আরো পরিষ্কার হবে।

৯. এবার ডবল ক্লিক করুন ডানের Add Name বোতামে। কোড এডিটরে বোতামের ক্লিক ইভেন্ট সাব দেখাবে। এই সাবের ভেতর লিখুন:

Private Sub btnMkt_Click(sender As Object, e As EventArgs) Handles btnMkt.Click

```
Dim MktPosition As String = ""
AddName("Marketing", MktPosition)
txtMkt.Text = txtMkt.Text & MktPosition
```

End Sub

ব্যাখ্যা: এই বোতামের কোড প্রায় একই। পার্থক্য হলো মডিউলের AddName প্রসিডিউরে "Marketing" পাটিয়ে দেয় এবং আপডেট করে txtMkt টেক্সট বাক্স।

১০. এবার ডবল ক্লিক করুন Quit বোতামে। এর ক্লিক ইভেন্টে লিখুন:

This sub quits the application

Private Sub btnQuit_Click(sender As Object, e As EventArgs) Handles btnQuit.Click

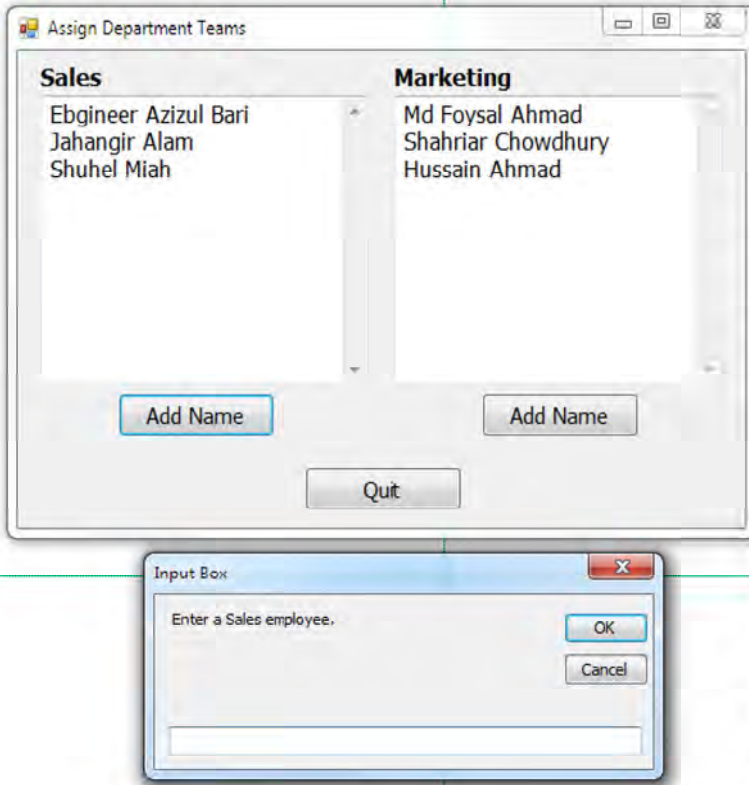
```
End
```

End Sub

১১. এবার Start মেন্যুতে ক্লিক করে প্রোগ্রাম পরিচালনা করুন। উভয় Add Name বোতামে ক্লিক করে দেখুন। পরের পৃষ্ঠায় প্রোগ্রাম চলাকালীন চিত্র দেখিয়েছি। এ উদাহরণ দ্বারা এটাই দেখানো হলো যে, একটি মাত্র সাব প্রসিডিউর ব্যবহার করে দুই বা ততোধিক ইনপুট কাজ করা যায়। এই মৌলিক টেকনিক কাজে লাগিয়ে বার বার ব্যবহারযোগ্য শক্তিশালী সাব এবং (পরবর্তীতে আলোচ্য) ফাংশন সৃষ্টি করতে পারেন আপনার প্রোগ্রামের জন্য।

৮ম পরিচ্ছেদ: *More on Subs & Functions*

সাব ও ফাংশন : *Subs & Functions*



প্রত্যেকবার Add Name বোতামে টিপ দিলেই নিচের ইনপুট বক্স ডায়ালগ আসবে এবং যে নাম লিখবেন তা টেক্সট বক্সে প্রদর্শিতও হবে। সুতরাং আপনি কোম্পানির সেল্‌স ও মার্কেটিং কর্মকর্তাদের একটি নামের লিস্ট বানাতে সক্ষম হলেন। হয়তো ভাবছেন এটা সংরক্ষণ করা যায় কিভাবে? এ ব্যাপারে ফাইল ম্যানেজমেন্টের উপর আলোচনাকালে ব্যাখ্যা আসবে। আপাতত এটাই জানা উদ্দেশ্য যে, একই প্রসিডিউর বার বার ব্যবহার করা সম্ভব।

Writing Function Procedure - ফাংশন প্রসিডিউর লেখা

আমরা জানি ফাংশন প্রসিডিউর হলো একগুচ্ছ ভিজুয়াল বেসিক.নেট কোড যা লেখা হয় Function এবং End Function এর মাঝখানে। এই কোডগুলোর কাজ হলো কোনো অর্থপূর্ণ কর্ম সম্পাদন। যেমন: টেক্সট প্রসেস করা, ইনপুট নিয়ন্ত্রণ কিংবা কোনো মান বেরকরণ ইত্যাদি

কাজে লাগাতে ফাংশন কল করতে হয়। ফাংশনে প্রয়োজনে আর্গুমেন্ট বা প্যারামিটার যুক্ত করা যায়। কলের সময় ব্র্যাকেটের ভেতর এসব প্যারামিটারের নাম দিতে হবে। মূলত কম্পিউটারে অসংখ্য রেডি-মেইড ফাংশন প্রসিডিউর আছে। প্রত্যেক এ্যাপলিকেশনই প্রয়োজনমত এসব ফাংশন ব্যবহার করে থাকে। আমরাও ব্যবহার করেছি। যেমন: MsgBox, InputBox ইত্যাদি। নিজের তৈরি ফাংশন ও এসব সিস্টেম ফাংশনে মূলত তেমন কোনো পার্থক্য নেই। ফাংশনের আরেকটি বৈশিষ্ট্য হলো এগুলো যদি আমরা আলাদা মডিউলে ডিফাইন করি তাহলে সমগ্র প্রোগ্রামব্যাপী তা ব্যবহার করতে পারি। একই কথা সাব প্রসিডিউরের ক্ষেত্রেও সত্য।

ফাংশনের আরেকটি উদাহরণ উপস্থাপনের পূর্বে তিনটি ব্যাপার পরিষ্কার করা প্রয়োজন বোধ করছি। যে কোনো ফাংশনের সিনটেক্সটে যা থাকা জরুরী তার মধ্যে এ তিনটি গুরুত্বপূর্ণ: ১. ফাংশনের নাম, ২. রিটার্ন টাইপ (যদি না থাকে তাহলে ভিজুয়াল বেসিক 'অবজেক্ট' টাইপ রিটার্ন করবে) ও ৩. ফাংশন স্টেটমেন্টস্। যেমন: নিচের ফাংশন দু'টি টেক্সট মান হিসাব করে যোগফল রিটার্ন করে। রিটার্ন মান হবে সিঙ্গুল ডাটা টাইপ।

Function TotalTax(ByVal Cost as Single) As Single

Dim StateTax, CityTax As Single

StateTax = Cost * 0.05 'State tax is 5% 'value of cost comes from calling line

CityTax = Cost * 0.015 'City tax is 1.5%

TotalTax = StateTax + CityTax 'to return value function name is used.

[Return StateTax + CityTax] 'রিটার্ন কিওয়ার্ড দ্বারাও মান ফেরৎ পাঠানো যায়।

End Function

কলিং: উক্ত ফাংশন কল করার উদাহরণ: TextBox1.Text = TotalText(400)।

লক্ষ করুন: ফাংশন নাম দ্বারা কিংবা **Return** কিওয়ার্ড দ্বারা মান রিটার্ন করা যায়। তবে উভয়টা ব্যবহার করা যাবে না।

যাক, এখন আমরা একটি নতুন প্রজেক্ট করবো এবং ফাংশনের ব্যবহার দেখাবো।

৯ম পরিচ্ছেদ: *Ready-made functions and methods*



বানানো ফাংশন ও মেথড: *Ready-made functions & Methods*

এ পরিচ্ছেদে আমরা বানানো কিছু ফাংশন ও মেথড নিয়ে উদাহরণসহ আলোচনা করবো। ভিজুয়াল বেসিক.নেট এর সৃষ্টিকারীরা প্রোগ্রামারদের সুবিধার জন্য প্রচুর ফাংশন ও মেথড তৈরি করে সিস্টেমে যুক্ত করে দিয়েছেন। এগুলো বিভিন্ন নাম ও নিয়মে ব্যবহার করা যায়। প্রথমে আমরা এরূপ কিছু ফাংশন উদাহরণসহ নিম্নে উপস্থাপন করছি।

ক. String Functions - স্ট্রিং ফাংশন -লেখা সম্পর্কিত ফাংশন

১. Split - মনে করুন: 'বাগানে অনেক আকর্ষণীয় ফুল ফুটেছে।' এ বাক্যটির প্রত্যেক শব্দ আলাদা করবেন। কোডিংয়ের মাধ্যমে এরূপ করতে চাইলে আপনি ব্যবহার করতে পারেন Split ফাংশনটি।

উদাহরণ:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    TextBox1.Text = "" 'clean text box
    Dim A_String As String = "evMvϣb AϣbK AvKl©Yxq dzj dzϣUϣQ!"
    'split the string based on spaces and put them in an Array variable
    Dim Words As String() = A_String.Split(New Char() {" "})
    'Display words in a text box in separate lines
    For A = 0 To Words.Length - 1 'Words.Length gives number of words in Array
        TextBox1.Text = TextBox1.Text & Words(A) & vbCrLf
    Next
End Sub
```

লক্ষ করুন: Split ফাংশন দিয়ে লেখার মধ্যে মোট ক'টি শব্দ আছে তা সহজে বের করা যায়। যেমন:

```
Dim Words As String() = A_String.Split(New Char() {" "})
MsgBox(Words.Length) 'this will show number of words
```

২. Join -আলাদা স্ট্রিং ইচ্ছে মতো একত্রিত করা যায় এই ফাংশন দিয়ে। যেমন:

'Join function at work

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    Dim An_Array(4) As String 'declare an array of 5 words
    An_Array(0) = "evMvϣb" : An_Array(1) = "AϣbK" : An_Array(2) = "AvKl©Yxq"
    An_Array(3) = "dzj" : An_Array(4) = "dzϣUϣQ!" 'put words into them
    Dim JoinText As String = String.Join(" ", An_Array) 'using space as delimiter
    TextBox1.Text = JoinText 'display result
End Sub
```

৩. Compare -দু'টি স্ট্রিং এর মধ্যে পার্থক্য নির্ণয় সময় সময় জরুরী হয়ে ওঠে। ভিজুয়াল বেসিক এ ফাংশন তৈরি করেছে পার্থক্য নির্ণয়ের জন্য। যেমন:

```
Dim StA As String = "a" : Dim StB As String = "b"
Dim C As Integer = String.Compare(StA, StB) : MsgBox(C) 'gives -1
C = String.CompareOrdinal(StB, StA) : MsgBox(C) 'gives 1
C = StB.CompareTo(StA) : MsgBox(C) 'gives 1
C = "X".CompareTo("X") : MsgBox(C) 'gives 0
```

এখানে: -১ অর্থ প্রথমটি ছোট, ১ অর্থ দ্বিতীয়টি বড় এবং ০ অর্থ উভয়টি সমান। যে কোনো শব্দ বা বাক্যও সমান কি না এই ফাংশন দ্বারা পরীক্ষা করা যায়। পরের পৃষ্ঠায় আরেকটি উদাহরণ দেখুন।

৯ম পরিচ্ছেদ: *Ready-made functions and methods*

বানানো ফাংশন ও মেথড: *Ready-made functions & Methods*

```
Dim Word1 As String = "আসমান" : Dim Word2 As String = "প্রথম"
'comapre
If String.Compare(Word1, Word2, False) = 0 Then
    MsgBox("String Match")
Else
    MsgBox("String does not Match") 'this message appears
End If
```

৪. Insert -এই ফাংশনের কাজ হলো স্ট্রিং এর ক্যারাক্টার পজিশনে প্রদত্ত স্ট্রিং ঢুকানো। যেমন:

'Insert the word [and].

```
Dim names As String = "Romeo Juliet"
Dim Shakespeare As String = names.Insert(6, "and ") 'and ' inserted in the space (character 'position
6: result: "Romeo and Juliet"
```

৫. Length -এই ফাংশনের কাজ হলো স্ট্রিং এর ভেতর ক'টি ক্যারাক্টার আছে তা নির্ণয় করা: যেমন:

```
Dim value As String = "dotnet"
Dim length As Integer = value.Length 'gives length = 6
value += ".aziz" 'now value is 'dotnet.aziz'
length = value.Length 'gives length = 11
```

৬. Remove -এটা দ্বারা কোনো স্ট্রিং এর প্রদত্ত পজিশন থেকে বাকি সবগুলো মুছে ফেলা যায়। যেমন:

'Completely incorrect input string.

```
Dim value As String = "Engineer Azizul Bari wrote 'Visual Basic.Net Programming'. and.."
'We need to remove 'and..' part, eg. everything after last SPACE
Dim index As Integer = value.LastIndexOf(" ")
'Remove everything starting at that position.
value = value.Remove(index)
TextBox1.Text = value
```

৭. Replace -এ ফাংশনের কাজ হলো যে কোনো স্ট্রিং - যেমন কোনো ডকুমেন্টের টেক্সটে একই শব্দ বা লাইন ইত্যাদি যতোবারই থাকুন না কেনো, তা এক সঙ্গে বদলানো। যেমন:

```
Dim FstString As String = "আমরা এই লেখার তাই শব্দ বদলাতে চাই। তাই, তাই"
```

ঋংঃঝঃৎরহম = ঋংঃঝঃৎরহম.জবঢ়মধপব("তাই", "")

MsgBox(FstString) 'gives 'আমরা এই লেখার শব্দ বদলাতে চাই।'

৮. Substring -এটা একটি সচরাচর ব্যবহৃত ফাংশন। এর কাজ হলো যে কোনো লেখা থেকে প্রদত্ত অক্ষরসমূহ সরিয়ে নিয়ে প্রদর্শন করা। যেমন:

'Use this string literal for the demonstration.

```
Dim literal As String = "Our Our country is Bangladesh." -আমরা 'Our' শব্দটি বের করবো।
```

```
Dim substring As String = literal.Substring(0, 4) 'first word
```

MsgBox(substring) 'will show - Our country is Bangladesh.

৯ম পরিচ্ছেদ: *Ready-made functions and methods*

বানানো ফাংশন ও মেথড: *Ready-made functions & Methods*

৯. Contains -এটা দ্বারা জানা যায় একটি স্ট্রিং এর মধ্যে আরেকটি স্ট্রিং আছে কি না। ফলাফল True বা False।
যেমন:

```
Dim literal As String = "লেখা থেকে প্রথম শব্দটি বের করবো।" : Dim newString As Boolean =  
literal.Contains("প্রথম")
```

```
MsgBox("প্রথম: " & newString) 'result True : Dim newStringt As Boolean = literal.Contains  
("আসমান") : MsgBox("আসমান: " & newStringt) 'result False
```

১০. InStr -এই ফাংশন দিয়ে একটি স্ট্রিং এর ভেতর আরো একটি স্ট্রিং আছে কি না পরীক্ষা করা যায়। যেমন:

```
Dim literal As String = "লেখা থেকে প্রথম শব্দটি বের করবো।"  
Dim isString As Integer = InStr(literal, "বের", CompareMethod.Text)  
MsgBox(isString) 'returns 24 -found at character position 24  
isString = InStr(literal, "আমার", CompareMethod.Binary)  
MsgBox(isString) 'retuns 0 -string not found
```

১১. Mid -এটি দিয়ে স্ট্রিং এর ভেতর থেকে কাঙ্ক্ষিত স্ট্রিং বের করা যায়। যেমন:

```
Dim literal As String = "লেখা থেকে প্রথম শব্দটি বের করবো।"  
Dim NString As String = Mid(literal, 6, Len(literal))  
MsgBox(NString) 'result- থেকে প্রথম শব্দটি বের করবো।
```

লক্ষ করুন: Length ফাংশন ও উপরে ব্যবহৃত Len ফাংশন এর কাজ একই। প্রথমটি ভিজুয়াল বেসিক-কে ভিজুয়াল বেসিক.নেট -এ উন্নীতকরণকালে সংযুক্ত করা হয়েছে। তবে উভয়টিই কার্যকরী।

১২. একটিমাত্র ক্যারাক্টার বারবার লেখা: নিচের কোডটি এ কাজ করে।

```
Dim txt As New String("স"c, 5)  
MsgBox(txt) -ফলাফল -সসসসস
```

১৩. ToUpper & ToLower -এ ফাংশন দ্বারা দু'টো শুধুমাত্র ইংরেজি অক্ষরসমূহ বড় এবং ছোট করা যায়। যেমন:

```
Dim word As String = "ViSuAL bASiC"  
MsgBox(word.ToUpper) 'gives VISUAL BASIC  
MsgBox(word.ToLower) 'gives visual basic
```

নোট: বাংলা ইউনিকোড লেখা হলে উভয় ফাংশন কাজ করে না। লেখা অপরিবর্তিত থাকবে।

১৪. স্ট্রিং -এর সকল অক্ষর আলাদা করে বের করা: নিচের কোডটুকু এ কাজ করবে। লক্ষণীয়- স্পেইস বা শূন্যস্থানকেও কোড ক্যারাক্টার হিসাবে সনাক্ত করে।

```
Dim AworD As String = "I am a programmer."  
For i As Integer = 0 To AworD.Length - 1 'Or 0 to len(AworD) -1  
MsgBox(AworD.Chars(i)) 'shows each character of the sentence  
Next
```


৯ম পরিচ্ছেদ: *Ready-made functions and methods*

বানানো ফাংশন ও মেথড: *Ready-made functions & Methods*

১৫. IsDigit -সময় সময় আমাদের জানতে হয় ক্যারাক্টার সংখ্যা [0-৯] কি না। এ ফাংশন দ্বারা তা জানা যায়। যেমন:

```
Dim EntTx As String = TextBox1.Text
If EntTx <> "" Then
    MsgBox(Char.IsDigit(EntTx.Chars(2)))
    'যদি তৃতীয় অক্ষর সংখ্যা হয়- ফলাফল True। অন্যথায় False হবে।
End If
```

অনুরূপ আরো দু'টি ফাংশন আছে: (ক) IsNumeric -এটা পুরো লেখাটি সংখ্যা (ডেসিমেলসহ) কি না দেখবে। (খ) IsNumer(Char, Character Position) -এটা ঠিক IsNumeric-এর মতো, তবে চিহ্নিত একটিমাত্র ক্যারাক্টার ইউনিকোড হলেও সনাক্ত করবে। যেমন বাংলা ইউনিকোড ক্যারাক্টার ৯ এ ফাংশন সংখ্যা হিসাবেই সনাক্ত করবে।

দৃষ্টান্ত:

```
Dim AString As String = "345.03"
If IsNumeric(AString) = True Then MsgBox("Yes") 'result Yes because whole string is a double
If Char.IsNumber(AString, 3) = True Then
    MsgBox("Yes")
Else
    MsgBox("No") 'result No because . isn't a number.
End If
AString = "৩৯৫.০৩" 'বাংলা ইউনিকোড সংখ্যা
If IsNumeric(AString) = True Then
    MsgBox("Yes")
Else
    MsgBox("No") 'result No because IsNumeric doesn't recognize unicode
End If
If Char.IsNumber(AString, 1) = True Then
    MsgBox("Yes") 'result Yes because index 1 = 9 [Remember: index always 0 based]
Else
    MsgBox("No")
End If
```

১৬. Trim, LTrim, RTrim -এ তিনটি ফাংশন যথাক্রমে লেখার উভয় দিকের, বায়ের ও ডানের শূন্যস্থান [স্পেস] মুছে ফেলে। যেমন:

```
Dim AString As String = " 345.03 is a number. "
AString = (Trim(AString)) 'both sides space erased
AString = (LTrim(AString)) 'left side space erased
AString = (RTrim(AString)) 'right side space erased
TextBox1.Text = AString 'Clean text: '345.03 is a number.'
```

খ. Conversion Functions - কনভারশন ফাংশন -রূপান্তর সম্পর্কিত ফাংশন

১. Asc -এই ফাংশন দিয়ে যে কোনো ক্যারাক্টারের সংখ্যা কোড জানা যায়। যেমন:

```
Dim codeInt As Integer
'The following line of code sets codeInt to 65.
codeInt = Asc("A")
'The following line of code sets codeInt to 97.
codeInt = Asc("a")
'The following line of code sets codeInt to 65.
codeInt = Asc("Apple")
```

৯ম পরিচ্ছেদ: *Ready-made functions and methods*

বানানো ফাংশন ও মেথড: *Ready-made functions & Methods*

২. AscW -এই ফাংশন দিয়ে যে কোনো ইউনিকোড ক্যারাক্টারের সংখ্যা কোড জানা যায়। কোড সংখ্যার ব্যাপ্তি হলো ০ থেকে ৬৫৫৩৫। উদাহরণ:

AscW("প") 'returns 2474

AscW("ক") 'returns 2453

৩. Chr -এ ফাংশন সংখ্যা কোড থেকে ক্যারাক্টার বের করে। যেমন:

Chr -returns character

Dim associatedChar As Char

' Returns "A".

associatedChar = Chr(65)

' Returns "a".

associatedChar = Chr(97)

' Returns ">".

associatedChar = Chr(62)

' Returns "%".

associatedChar = Chr(37)

৪. ChrW -এ ফাংশন সংখ্যা কোড থেকে ইউনিকোড ক্যারাক্টার বের করে। যেমন:

ChrW(2474) returns 'প'

ChrW(2453) returns 'ক'

৫. Format -এটি সাধারণত তারিখ ফরম্যাট করার জন্য ব্যবহৃত হয়। যেমন:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim TestDateTime As Date = #1/27/2001 5:04:23 PM#
    Dim TestStr As String
    TestStr = Format(Now(), "Long Time") ' Returns current system time in the system-defined long time format.
    TextBox1.Text += TestStr & vbCrLf ' Also returns current system date in the system-defined long date format,
    using the single letter code for the format.
    TestStr = Format(Now(), "D")
    TextBox1.Text += TestStr & vbCrLf
    TestStr = Format(TestDateTime, "h:m:s") ' Returns the value of TestDateTime in user-defined date/time
    formats. Returns "5:4:23".
    TextBox1.Text += TestStr & vbCrLf
    TestStr = Format(TestDateTime, "hh:mm:ss tt") ' Returns "05:04:23 PM".
    TextBox1.Text += TestStr & vbCrLf
    TestStr = Format(TestDateTime, "dddd, MMM d yyyy") ' Returns
    "Saturday, Jan 27 2001".
    TextBox1.Text += TestStr & vbCrLf
    TestStr = Format(TestDateTime, "HH:mm:ss") ' Returns "17:04:23".
    TextBox1.Text += TestStr & vbCrLf
    TestStr = Format(23) ' Returns "23"
    TextBox1.Text += TestStr
End Sub
```

বায়ের প্রোগ্রাম রান করার ফলাফল: 12:50:54 PM Wednesday, August 10, 2016 5:4:23 05:04:23 PM Saturday, Jan 27 2001 17:04:23 23
--

৯ম পরিচ্ছেদ: *Ready-made functions and methods*

বানানো ফাংশন ও মেথড: *Ready-made functions & Methods*

অবশ্য ফাংশন দ্বারা ব্যবহারকারীর ইচ্ছেমাফিক কাজও করা যায়। যেমন নিচের কোড দ্বারা সংখ্যা ফরম্যাট করা যাবে:

```
Dim TestStr as String
```

```
TestStr = Format(54590125.4, "###,##,##0.00") 'Results 54,590,125.40
```

গ. **Math Functions** - ম্যাথ ফাংশন - গাণিতিক ফাংশন। এসব ফাংশনের নামের পূর্বে **Math.** শব্দ লিখতে হবে।

১. **Math.Abs** -এ ফাংশন যে কোনো সংখ্যাকে পজিটিভ সংখ্যায় পরিণত করে। যেমন:

```
Math.Abs(-123) -returns 123
```

২. **Math.Acos** -এটা কোণের Cosine সংখ্যা থেকে ডিগ্রী রেডিয়ান দেবে। যেমন:

```
Math.Acos(0.9568) -returns 0.295 radians
```

৩. **Math.Asin** -এটা কোণের Sine সংখ্যা থেকে ডিগ্রী রেডিয়ান দেবে। যেমন:

```
Math.Asin(0.145268) -returns 0.14578 radians
```

অন্যান্য হলো: Atan [Arc Tan], Cos [Cosine], Cosh [Hyperbolic Cosine], Sin [Sine], Sinh [Hyperbolic Sine], Tan [Tangent], Tanh [Hyperbolic Tan], Exp [Exponential - e^y to the power of y], Pow [x^y Value], Log [Logarithm -base 2], Log10 [Logarithm -base 10], Sqrt [Square Root], Sign [Integer value with + or - Sign], Min [Smaller of two numbers], Max [Larger of two numbers]

আরো কিছু ম্যাথ ফাংশন হলো:

(ক) **Math.BigMul**(1256, 2154) -returns 2705424

(খ) **Math.IEEERemainder**(45,7) -returns (45/7) remainder 3

(গ) **Math.Round**(45.8) -returns 46; **Math.Round**(45.81255, 3) -returns 45.823

(ঘ) **Math.Truncate**(45.81255) -returns 45

ঘ. **Type Conversion Functions** -টাইপ (ভেরিয়েবল ধরন) রূপান্তর ফাংশন

১. Integer > < Boolean - উদাহরণ:

```
Dim a, b, c As Integer
```

```
Dim check As Boolean
```

```
a = 5 : b = 5 ' The following line of code sets check to True.
```

```
check = CBool(a = b) 'check is True
```

```
c = 0 ' The following line of code sets check to False.
```

```
check = CBool(c) 'check is False
```

২. Double > < Byte - উদাহরণ:

```
Dim aDouble As Double : Dim aByte As Byte
```

```
aDouble = 125.5678 ' The following line of code sets aByte to 126.
```

```
aByte = CByte(aDouble)
```

৯ম পরিচ্ছেদ: *Ready-made functions and methods*

বানানো ফাংশন ও মেথড: *Ready-made functions & Methods*

৩. String > < Char, CChar - উদাহরণ:

```
Dim aString As String
Dim aChar As Char
aString = "BCD" ' CChar converts only the first character of the string.
aChar = CChar(aString) 'sets aChar to "B".
```

৪. String > < Integer, Char - উদাহরণ:

```
Dim someDigits As String
Dim codePoint As Integer
Dim thisChar As Char
someDigits = InputBox("Enter code point of character:") 'ask the user
codePoint = CInt(someDigits) 'convert it to integer and equate ti declared integer type
' The following line of code sets thisChar to the Char value of codePoint.
thisChar = ChrW(codePoint)
```

৫. String > < To Date and Time -উদাহরণ:

```
Dim aDateString, aTimeString As String
Dim aDate, aTime As Date
aDateString = "February 12, 1969"
aTimeString = "4:35:47 PM" ' The following line of code sets aDate to a Date value.
aDate = CDate(aDateString) ' The following line of code sets aTime to Date value.
aTime = CDate(aTimeString)
```

৬. Double > < To Integer -উদাহরণ:

```
Dim aDbl As Double
Dim anInt As Integer
aDbl = 2345.5678 ' The following line of code sets anInt to 2346.
anInt = CInt(aDbl)
```

৭. Double > < To Object -ভিজুয়াল বেসিক.নেটে Object -একটি টাইপ ভেরিয়েবল। উদাহরণ:

```
Dim aDouble As Double
Dim anObject As Object
aDouble = 2.7182818284 ' The following line of code sets anObject to a pointer to
aDouble.
anObject = CObj(aDouble)
```

দ্র: উপরের কোড অংশের কমেণ্টে pointer-শব্দটি ব্যবহার করেছি। pointer কি? প্রোগ্রামিং ভাষায় pointer হলো: প্রোগ্রামিং ভাষার বস্তু [Object], যার মান [Value] ইঙ্গিতে নির্দেশ করে [Points to] কম্পিউটারের অন্যত্র মেমোরি ঠিকানায় সংরক্ষিত মানের দিকে। আরো বুঝতে আমরা গ্রন্থের 'সূচিপত্রের' কথা ভাবতে পারি। সূচিতে সংশ্লিষ্ট বিষয়ের উপর আলোচনার পৃষ্ঠা সংখ্যা থাকে। অন্যকথায়, বইয়ের সূচিতে দেওয়া এ পৃষ্ঠা নম্বার হলো একটি পয়েন্টার যা আমাদেরকে জানিয়ে দেয় কোন্ পৃষ্ঠায় তথ্যাদি সংরক্ষিত [লিখিত] হয়েছে। পয়েন্টার ব্যবহারের মাধ্যমে কম্পিউটিং গতি বৃদ্ধি পায়। কারণ, একই ঠিকানায় সংরক্ষিত তথ্য বিভিন্ন কোড লাইনে ব্যবহৃত হয়ে থাকে।

৯ম পরিচ্ছেদ: *Ready-made functions and methods*

বানানো ফাংশন ও মেথড: *Ready-made functions & Methods*

৮. **String > < Char, CChar** - উদাহরণ:

```
Dim aString As String
Dim aChar As Char
aString = "BCD" ' CChar converts only the first character of the string.
aChar = CChar(aString) 'sets aChar to "B".
```

৮. **Double > < Byte, Short** - উদাহরণ:

```
Dim aDouble As Double
Dim anSByte As SByte
aDouble = 39.501 ' The following line of code sets anSByte to 40.
anSByte = CSByte(aDouble)
Dim aByte As Byte
Dim aShort As Short
aByte = 100 ' The following line of code sets aShort to 100.
aShort = CShort(aByte)
```

Byte data type -কম্পিউটারের মেমোরিতে সংরক্ষিত সর্বাধিক স্বল্পমাত্রার সাংখ্যিক তথ্যকে বলে বিট [Bit]। একেক বিটকে বিজ্ঞানীরা দু'টি মাত্র সাংখ্যিক কোড দ্বারা উপস্থাপন করেন। এগুলো হলো: ০ এবং ১। এ দু'টোর সমন্বয়ে ঘটিত ৮টি বিট হলো একেকটি বাইট [Byte]। সুতরাং ১ বাইট মানে ৮ বিট তথ্য।

৯. **Double > < String** - উদাহরণ:

```
Dim aDouble As Double
Dim aString As String
aDouble = 437.324 ' The following line of code sets aString to "437.324".
aString = CStr(aDouble)
```

১০. **Double > < String** - উদাহরণ:

```
Dim testNumber As Long = 1000 ' The following line of code sets testNewType to 1000.0.
Dim testNewType As Single = CType(testNumber, Single)
```

১১. **Properties** -ভিজুয়াল বেসিকে প্রোপার্টি

১. **DateString** -কম্পিউটারের সেটিং মুতাবিক বর্তমান তারিখ ও সময়। ব্যবহার:

```
MsgBox("The current date is " & DateString) 'returns 08-11-2016
MsgBox("The formatted date is " & Format(Now, "dddd, d MMM yyyy")) 'returns
Thursday, 11 August 2016.
```

২. **Now** -এখনকার তারিখ ও সময়। ব্যবহার:

```
Dim ThisMoment As Date ' The following statement calls the Get procedure of the Visual
Basic Now property.
```

```
ThisMoment = Now : MsgBox(ThisMoment) 'retuns 8/11/2016 12:44:03 PM
```

৯ম পরিচ্ছেদ: *Ready-made functions and methods*

বানানো ফাংশন ও মেথড: *Ready-made functions & Methods*

৩. **ScriptEngine** -বর্তমান রান-টাইমের ওপর তথ্য। উদাহরণ:

MsgBox(ScriptEngine) 'returns- VB

৪. **ScriptEngineBuildVersion** -বর্তমান রান-টাইমের বিল্ড ভার্সনের ওপর তথ্য। উদাহরণ:

MsgBox(ScriptEngineBuildVersion) 'returns- 50709

৫. **ScriptEngineMajorVersion** -বর্তমান রান-টাইমের মেজর ভার্সনের ওপর তথ্য। উদাহরণ:

MsgBox(ScriptEngineMajorVersion) 'returns 11

৬. **ScriptEngineMinorVersion** -বর্তমান রান-টাইমের মাইনোর ভার্সনের ওপর তথ্য। উদাহরণ:

MsgBox(ScriptEngineMinorVersion) 'returns 0

৭. **TimeOfDay** -দিনের বর্তমান সময়ের ওপর তারিখ ভ্যালু সেট বা রিটার্ন করে। উদাহরণ:

Dim currentTime As Date

currentTime = TimeOfDay 'returns 1:10:19 PM

৭. **Timer** -আগের মধ্যরাত থেকে বর্তমান পর্যন্ত কতো সেকেন্ড অতিবাহিত হয়েছে তার একটি মাত্রা। উদাহরণ:

```
Public Sub waitFiveSeconds()
```

```
    If TimeOfDay >= #11:59:55 PM# Then
```

```
        MsgBox("The current time is within 5 seconds of midnight" &
            vbCrLf & "The timer returns to 0.0 at midnight")
```

```
        Return
```

```
    End If
```

```
    Dim start, finish, totalTime As Double
```

```
    If (MsgBox("Press Yes to pause for 5 seconds", MsgBoxStyle.YesNo) = MsgBoxResult.Yes) Then
```

```
        start = Microsoft.VisualBasic.DateAndTime.Timer
```

```
        ' Set end time for 5-second duration.
```

```
        finish = start + 5.0
```

```
        Do While Microsoft.VisualBasic.DateAndTime.Timer < finish
```

```
            ' Do other processing while waiting for 5 seconds to elapse.
```

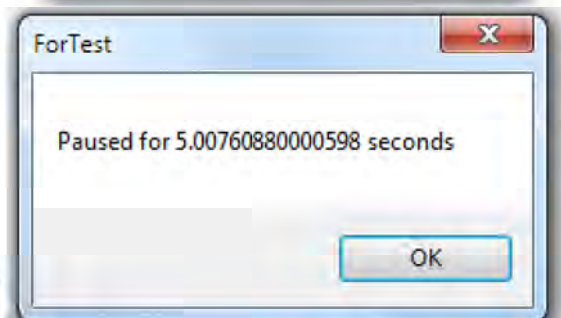
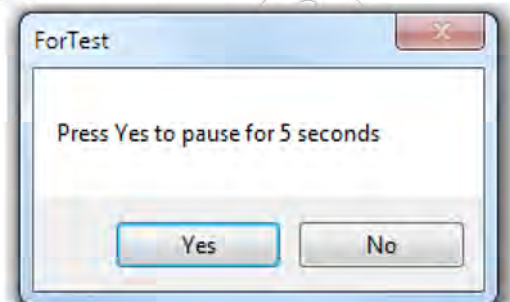
```
        Loop
```

```
        totalTime = Microsoft.VisualBasic.DateAndTime.Timer - start
```

```
        MsgBox("Paused for " & totalTime & " seconds")
```

```
    End If
```

```
End Sub
```



উপরের প্রোগ্রাম রানিংয়ের চিত্র।

১০ম পরিচ্ছেদ: *Object Oriented Programming [OOP]* -বস্তুভিত্তিক প্রোগ্রামিং



বস্তুভিত্তিক প্রোগ্রামিং: *Object Oriented Programming [OOP]*

শিক্ষার্থীরা অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং [OOP] কথাটি ইতোমধ্যে অন্যত্র দেখেছেন। এ সম্পর্কে বিস্তারিত আলোচনা জরুরী মনে করে এ পরিচ্ছেদ লেখার প্রয়াস। কম্পিউটার সফটওয়্যার সৃষ্টির জগতে OOP একটি আধুনিক ধারণা [concept]। ভিজুয়াল বেসিক.নেট এর পূর্বসূরী ছিলো শুধু ভিজুয়াল বেসিক। ভার্সন ৬.০ এর পর মাইক্রোসফট ভিজুয়াল বেসিক আর ডেভোলাপ করে নি। ভিজুয়াল বেসিক.নেট এর ওপরই মনোযোগ দিয়েছে। এর একটি অন্যতম কারণ, আধুনিক অন্যান্য ভাষা [যেমন c++, Java] হলো পূর্ণাঙ্গ OOP-নিয়ন্ত্রিত। মাইক্রোসফট ভিজুয়াল বেসিকের মতো নতুন একটি ভাষা তৈরি করতে সচেষ্ট হয় পৃথিবীব্যাপী অসংখ্য ভক্ত-অনুরক্ত প্রোগ্রামারদের সুবিধার্থে। সবাই চাচ্ছিলো একটি উন্নতমানের ভাষা যাতে সরলতা থাকবে- যেরূপ মূল ভিজুয়াল বেসিকে ছিলো, এসাথে আধুনিক অন্যান্য ভাষার সঙ্গে সামঞ্জস্যশীলতা নিশ্চিত করা। আর এরই ফলাফল স্বরূপ আমরা পেয়েছি পূর্ণাঙ্গ OOP-সহ মাইক্রোসফটের এই .নেট ভাষাটি। এই ভাষার আরেক বৈশিষ্ট্য হলো উইন্ডোজ অপারেটিং সিস্টেমের ফাংশন, মেথড ইত্যাদি সরাসরি কাজে লাগানোর সুযোগ। এ সুযোগের নামই হলো API- Application Program Interface। মূলত একটি কোডিং সিস্টেম যেখানে সফটওয়্যার তৈরিকারী নিজের কোড দ্বারা সিস্টেমের কোডের সঙ্গে যোগাযোগ সৃষ্টি করতে পারেন।

যাক, ফিরে আসি OOP প্রোগ্রামিং কনসেপ্টে। আমাদের এ জগতটি মূলত বস্তু বা অবজেক্টের তৈরি। একটি বাড়ি। এটা সার্বিকভাবে একটি বস্তু যা সৃষ্টি হয়েছে একাধিক বিভিন্ন অংশের সমন্বয়ে। যেমন: ইট, পাথর, বালি, দেওয়াল, ফ্লোর, কক্ষ, ছাদ, পিলার ইত্যাদি। এ-সবই বাড়ির অংশ বা প্রোপার্টি। অনুরূপ একটি বৃক্ষের মধ্যে বেশ ক'টি প্রোপার্টি চিহ্নিত করা যায়। যেমন: এর মূল, কাণ্ড, শাখা, প্রশাখা, পত্র, উচ্চতা, কাণ্ডের ব্যাস ইত্যাদি। কম্পিউটার বিজ্ঞানীরা এরূপ প্রাত্যহিক বস্তু ও এদের প্রোপার্টি গবেষণা করে OOP সৃষ্টি করেছেন প্রোগ্রামিং সুবিধার্থে।

যে কোনো প্রোগ্রামিং ভাষাকে OOP হিসেবে স্বীকৃতি পেতে হলে তার মধ্যে তিনটি মৌলিক জিনিস থাকতে হবে:

ক. **Encapsulation** -ঠিক যেরূপ ক্যাপসুলের ভেতর কিছু থাকে সেরূপ কোডগুলো বন্ধ অবস্থায় থাকা। অন্যকথায় বাইরের জগত থেকে গোপন রাখা।

খ. **Inheritance** -ঠিক যেরূপ কোনো প্রাণী নিজের বংশবিস্তার লাভে সমর্থ, সেরূপ নিজেকে কপি করার ক্ষমতা থাকা। অর্থাৎ উত্তরাধিকার ব্যবস্থা থাকতে হবে।

গ. **Polymorphism** -ঠিক একই দলভুক্ত বিভিন্ন বস্তুর মতো থাকতে হবে, একই দলভুক্ত একাধিক ভিন্ন বস্তু।

এই তিনটি বিষয়ের ওপর আরো কিছু ব্যাখ্যা তুলে ধরার প্রয়াস পাচ্ছি। এগুলো বুঝে এসে গেলে OOP সম্পর্কে ধারণাও আরো স্পষ্ট হয়ে ওঠবে।

ক. **Encapsulation**

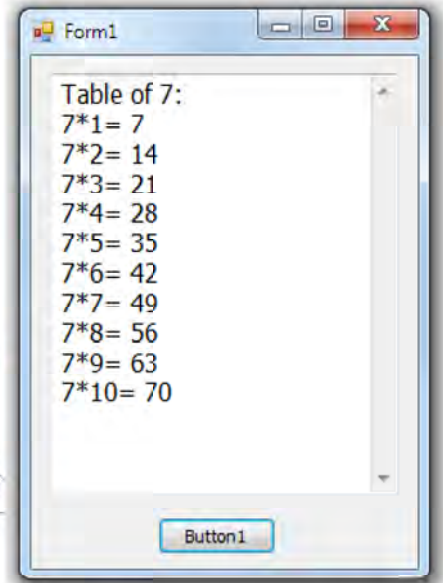
কোডিংয়ের ক্ষেত্রে বার বার ব্যবহারযোগ্য এবং ভিন্ন এ্যাপলিকেশনে বহনযোগ্য কিছু ইউনিট তৈরি করা যায়। এগুলোকে বলে আত্মনির্ভরশীল মডিউল [self-contained module]। এসব মডিউলে থাকে তথ্য প্রসেস করার বিভিন্ন ফাংশন। ব্যবহারকারী কর্তৃক নিরূপণকৃত তথ্য-ধরনকে [user-defined data type] বলে ক্লাস [class]। প্রত্যেক ক্লাসে থাকে তথ্য [data]। আরো থাকে এক বা একাধিক পদ্ধতি [method] যাদের মাধ্যমে তথ্যাদি রদবদল করা যায়। ক্যাপসুলের মতো আত্মনির্ভরশীল আবদ্ধ ক্লাসের ডাটা অংশকে বলে সময়ের পরিবর্তনশীল তথ্য বা ইন্সট্যান্ট ভেরিয়েবলস [instant variables]। আর ক্লাসের একটি ইন্সট্যান্ট হলো একটি বস্তু [object]। আমরা এগুলোকে পাবলিক লাইব্রেরি সিস্টেমের সঙ্গে তুলনা করতে পারি। ক্লাস হতে পারে লাইব্রেরির সদস্য এবং আমির ও সাকিয়া হতে পারে এ লাইব্রেরি ক্লাসের দু'টি ইন্সট্যান্ট [দু'টি বস্তু]। আশারাখি এ কথাগুলো পাঠ করে আপনি Encapsulation সম্পর্কে আরো কিছুটা স্পষ্ট ধারণা পেয়ে গেছেন। তবে বাস্তব ক্ষেত্রে একটি কোড উদাহরণ দেখুন। এতে বিষয়টি আরো স্পষ্ট হবে।

১০ম পরিচ্ছেদ: **Object Oriented Programming [OOP]** - বস্তুভিত্তিক প্রোগ্রামিং

বস্তুভিত্তিক প্রোগ্রামিং: **Object Oriented Programming [OOP]**

ভিজুয়াল বেসিক.নেটে **Encapsulation** ব্যবহারের উদাহরণ: একটি গুণের নামতা:

```
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        TextBox1.Text = "" 'clear text
        MySub() 'calling another sub
    End Sub
    'called from button clicked event sub
    Public Sub MySub()
        Dim obj As ClassA = New ClassA(7) 'create object of ClassA
        obj.table() 'calling member method by object [function call which is in classA]]
    End Sub
End Class
=====
'class declaration
Public Class ClassA 'ClassA starts
    Private Value As Integer 'private member variable
    Sub New(V As Integer) 'constructor with 1 argument
        Value = V
    End Sub
    Public Function table() 'member method to show member variable
        Dim Heading, TableV As String
        Heading = "Table of " & Value & ": " & vbCrLf 'heading text String with
return vbCrLf Character
        Form1.TextBox1.Text += Heading 'display heading in the TextBox
        For i As Integer = 1 To 10 Step 1 'calculate times table and display
            TableV = Value & "*" & i & "=" & Value * i & vbCrLf 'calculate and
put into variable
            Form1.TextBox1.Text += TableV 'display '+' means add text to
existing text.
        Next
        Return Nothing
    End Function
End Class 'classA ends
```



বায়ের প্রোগ্রামের ফলাফল।

খ. Inheritance

ক্লাসগুলো সৃষ্টি হয় হাইআর্কির [hierarchies] সূত্র ধরে। অর্থাৎ ক্রমাধিকারের মাধ্যমে এগুলো অস্তিত্বশীল হয়। ইনহেরিট্যান্স [Inheritance] বা উত্তরাধিকার থাকায় কোড কাঠামো ও মেথড একটি ক্লাস থেকে হাইআর্কির নিচ দিকে যেতে সক্ষম হয়। এর ফলে জটিল সিস্টেমেও কম প্রোগ্রামিংয়ের জরুরত হবে। অন্যকথায় ভিন্ন কোনো পদক্ষেপ হাইআর্কির নিম্নতম স্তরে যুক্ত করতে হলে পুরো হাইআর্কিকে অপরিবর্তিত রেখেই শুধুমাত্র ঐ পদক্ষেপটি তৈরি করে যুক্ত করা সম্ভব হবে। এই পদক্ষেপের বাকি সকলই উত্তরাধিকারসূত্রে পাওয়া যাবে [inherited]। এভাবে প্রস্তুতকৃত বস্তু পুনঃব্যবহারের সুযোগ থাকাটাকে অবজেক্ট প্রযুক্তির ক্ষেত্রে সর্বাধিক বড় সুবিধা হিসাবে ধারণা করা হয়।

একটি ক্লাস যখন অপর আরেকটি ক্লাস Inherit করে বা উত্তরাধিকারসূত্রে প্রাপ্ত হয় তখন একে বলে derived [উদ্ভূত] ক্লাস। অপরদিকে উদ্ভূত ক্লাস থেকে যে ক্লাস উত্তরাধিকারসূত্রে প্রাপ্ত হয় তাকে বলে Base Class [মূল ক্লাস]। কেউ কেউ অন্যান্য নামও ব্যবহার করেন, যেমন: প্যারেন্ট [অভিভাবক], চাইল্ড [সন্তান] ইত্যাদি। এবার একটি উদাহরণ দেখুন।

```
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        TextBox1.Text = "" 'clear text
        InVokes()
    End Sub
```


১০ম পরিচ্ছেদ: **Object Oriented Programming [OOP]** - বস্তুভিত্তিক প্রোগ্রামিং

বস্তুভিত্তিক প্রোগ্রামিং: **Object Oriented Programming [OOP]**

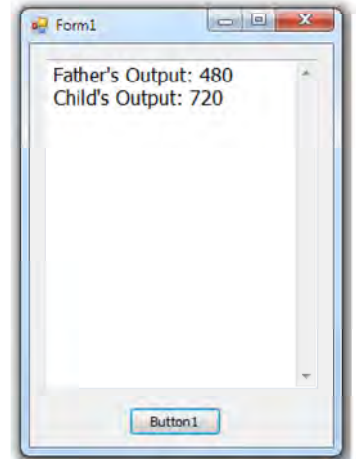
```

Sub InVokes() 'method- called from button's click event sub
    Dim obj As Child = New Child(5, 48)
    TextBox1.Text += "Father's Output: " & obj.FatherOutput & vbCrLf
    TextBox1.Text += "Child's Output: " & obj.ChildOutput
End Sub
End Class
'=====Base Class=====
Public Class Father 'start of base class Father
    Protected Resources As Integer
    Protected WorkInHours As Integer
    Public Sub New(r As Integer, w As Integer) 'constructor with 2 arguments
        Resources = r
        WorkInHours = w
    End Sub

    Public Function FatherOutput() As Integer 'method
        Dim Output As Integer = Resources * WorkInHours
        Return Output
    End Function
End Class
'=====Inherited Class=====
Public Class Child : Inherits Father 'derived class
    Private Resources As Integer
    Private WorkInHours As Integer
    Public Sub New(r As Integer, w As Integer) 'constructor with 2 arguments
        MyBase.New(10, 48)
        Me.Resources = r
        Me.WorkInHours = w
    End Sub

    Public Function ChildOutput() As Integer 'method
        Dim Output As Integer = (Me.Resources + MyBase.Resources) * WorkInHours
        Return Output
    End Function
End Class

```



বায়ের প্রোগ্রাম রানের ফলাফল

খ. Polymorphism

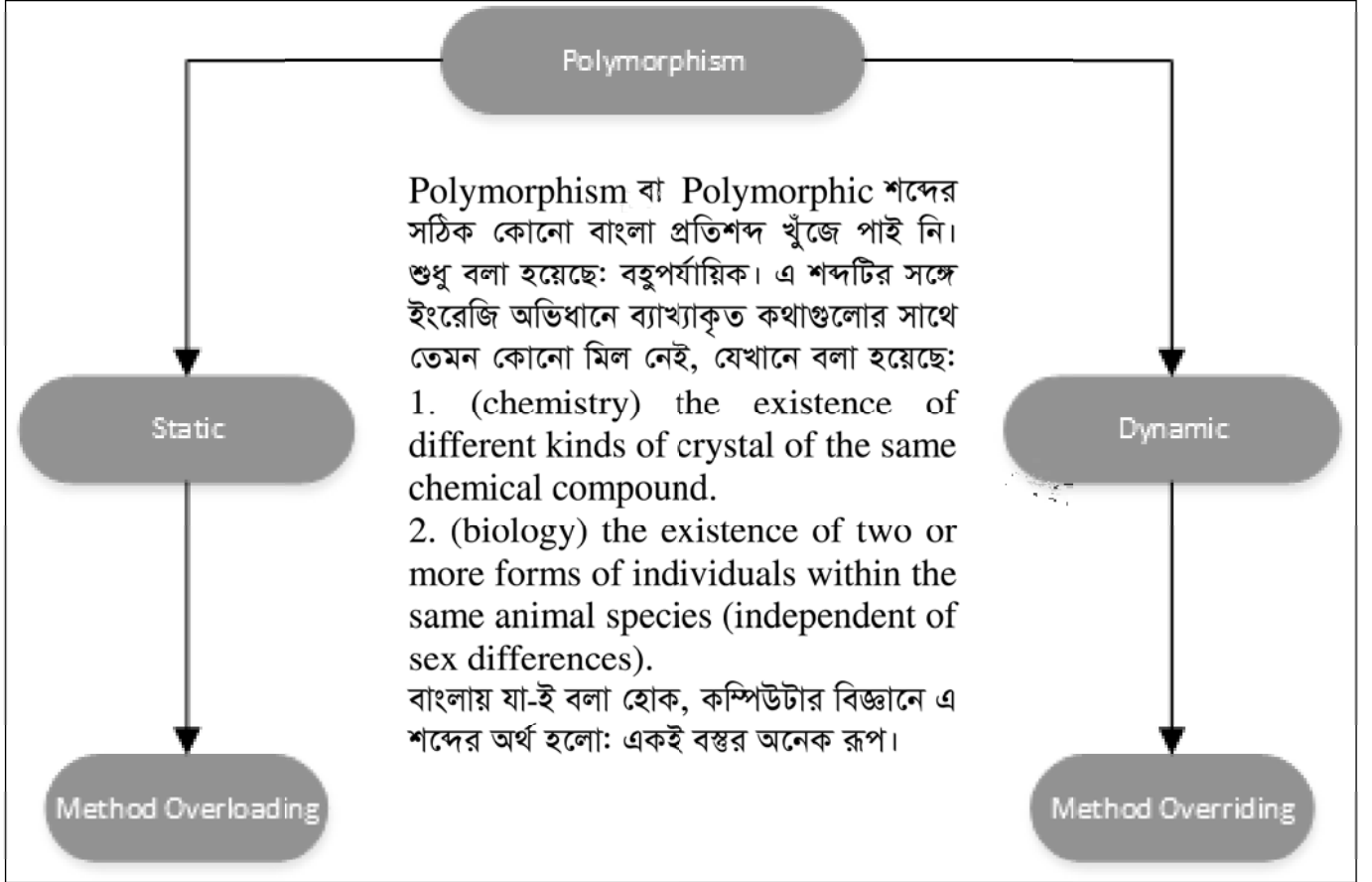
শব্দটি আসলে রসায়ন ও জীববিদ্যা থেকে ধার-করা। রসায়ন শাস্ত্রে Polymorphism-এর অর্থ হলো একই রসায়নভুক্ত কম্পাউন্ড [যুক্ত পদার্থ] দ্বারা সৃষ্ট বিভিন্ন ক্রিস্ট্যাল [ক্ষটিক]। জীববিদ্যায় এ শব্দের অর্থ হলো একই প্রজাতিভুক্ত বিভিন্ন প্রাণীর অস্তিত্ব যা মূলত লিঙ্গ নির্ভরশীলহীন।

অবজেক্ট অরিয়েন্টেড প্রোগ্রামিংয়ে বিমূর্ত [abstract] কোড কাঠামো ও মেথড [পদ্ধতি] তৈরি সম্ভব যাদের যথার্থ ধরন বা টাইপ রানটাইম [প্রোগ্রাম চলাকালীন অবস্থা] এর পূর্বে জানা যায় না। যেমন ধরুন, স্ক্রীনের কারসার মাউস চলাকালে তার আকার তীর [এ্যারো] থেকে বদলে রেখায় [লাইন] পরিণত হতে পারে। এটা নির্ভর করবে প্রোগ্রামের ভাবধারার ওপর। সুতরাং আপনি প্রোগ্রামার হিসেবে স্ক্রীনে মাউস চালানোর জন্য লিখবেন 'cursor' কন্ট্রোল [কোড ব্লক]। আর পলিমোরফিজম রানটাইমের সময় মাউসের আকার প্রয়োজনমত রদবদল করবে। এটা নতুন এই আকারকে সহজে সম্পৃক্ত করতেও সুযোগ সৃষ্টি করে দেবে। অন্যকথায় রসায়ন শাস্ত্রে যে রূপ একই রসায়নভুক্ত কম্পাউন্ড থেকে সৃষ্টি হতে পারে বিভিন্ন ক্রিস্ট্যাল এবং জীববিদ্যায় একই প্রজাতিভুক্ত বিভিন্ন প্রাণীর অস্তিত্ব সম্ভব- তদ্রূপ অবজেক্ট অরিয়েন্টেড প্রোগ্রামিংয়েও রান-টাইমের সময় একটি অবজেক্ট প্রয়োজনমত তার আকার, আয়তন, কর্ম ইত্যাদিও রদবদল করার ক্ষমতা রাখতে পারে। আর একেই বলে **Polymorphism**। কোডে পলিমোরফিজম হয় মেথড অভাররাইডিং ও মেথড অভারলোডিং এর মাধ্যমে। উদাহরণ দ্বারা বুঝা অনেকটা সহজ। [পরের পৃষ্ঠায় দেখুন]

১০ম পরিচ্ছেদ: **Object Oriented Programming [OOP]** -বস্তুভিত্তিক প্রোগ্রামিং

বস্তুভিত্তিক প্রোগ্রামিং: **Object Oriented Programming [OOP]**

Polymorphism এর দু'টি রূপ আছে: ১. Static Polymorphism [স্থিত পলিমোরফিজম] ও ২. Dynamic Polymorphism [বেগবান পলিমোরফিজম]। নিচের ব্লক চিত্র থেকে উভয়টির স্বরূপ সম্পর্কে অতিরিক্ত ধারণা পাওয়া যেতে পারে।



উপরের ডায়াগ্রাম থেকে স্পষ্ট, পলিমোরফিজমের স্টেটিক [স্থিত] শাখার মেথড [কোড পদ্ধতি] হলো অভারলোডিং [অতিরিক্ত বোঝাই করা]। অরপদিকে ডাইনামিক [বেগবান] শাখার মেথড হবে অভাররাইডিং [পদদলিতকরণ!]।

১. Static Polymorphism [স্থিত পলিমোরফিজম]: একটি উদাহরণ

একে প্রাথমিক বন্ধন বা Early Binding কিংবা ভাষা পরিবর্তনকাল অর্থাৎ Compile Time Polymorphism-ও বলে। অন্যকথায় সিদ্ধান্ত নেওয়া হয় কম্পাইল টাইমে। মেথড অভারলোডিং হলো কম্পাইল টাইমেই কম্পাইলার জানতে পারে কোন্ মেথড সম্পাদিত হবে। কম্পাইল টাইমে কম্পাইলার প্যারামিটার সংখ্যা এবং প্যারামিটারের ধরন বুঝে সিদ্ধান্ত নেয় কোন্ অভারলোডেড মেথড সম্পাদিত [execute] হবে। উদাহরণ কোডটি দেখে নিই।

Public Class Student 'class starts

'show method with 3 overloaded versions

Public Overloads Function show(name As String) As String 'method with 1 parameter

Return name

End Function

[বাকি কোড পরের পৃষ্ঠায়]

১০ম পরিচ্ছেদ: **Object Oriented Programming [OOP]** - বস্তুভিত্তিক প্রোগ্রামিং

বস্তুভিত্তিক প্রোগ্রামিং: **Object Oriented Programming [OOP]**

```
Public Overloads Function show(name As String, rollNo As String) As String 'method with 2 parameter
    Return (name + " " + rollNo)
End Function
```

'method with 3 parameter

```
Public Overloads Function show(name As String, rollNo As String, Section As String) As String
    Dim St As String = (name + " " + rollNo + " " + Section)
    Form1.TextBox2.Text = St 'display result
    Return Nothing
End Function
End Class
```

Public Class Form1

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
```

```
    TextBox1.Text = "" 'clear text
```

```
    TextBox2.Text = "" 'clear text
```

```
    CallSub()
End Sub
```

'called from button clicked event sub

```
Public Sub CallSub()
```

```
    Dim s As Student = New Student()
```

```
    TextBox1.Text += (s.show("Engineer Azizul Bari")) & vbCrLf 'call show method with 1 param
```

```
    'call show method with 3 param
```

```
    TextBox1.Text += (s.show("Engineer Azizul Bari", "0173249 8288", "Bangladesh")) & vbCrLf
```

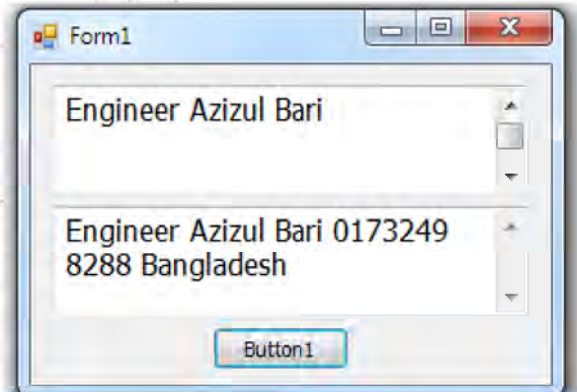
```
End Sub
```

End Class

২. Dynamic Polymorphism [বেগবান পলিমোরফিজম]:

একটি উদাহরণ

একে রান-টাইম (চলাকালীন) বা লেইট-বাইন্ডিং (দেরিতে-চলন্ত) পলিমোরফিজম-ও বলে। মেথড অভাররাইডিং এর মাধ্যমে এটা সম্পাদিত হয়। মেথড অভাররাইডিং মানে একই মেথডের বিভিন্ন রূপকে কল করা। কোন রূপকে কল করা হবে তা নির্ভর করবে প্রোগ্রাম চলাকালে বেগবান অবস্থায়। অন্যকথায়, রান-টাইম পলিমোরফিজমে কোন মেথড কল করা হবে তা নির্দিষ্ট হয় রেফারেন্স ভেরিয়েবল কর্তৃক পছন্দকৃত বস্তু থেকে। উদাহরণ থেকে এসব কথার সঠিক অর্থ হয়তো অনুধাবন হতে পারে।



উপরের প্রোগ্রাম পরিচালনার ফলাফল

Public Class Add 'base class starts

'Virtual method of Base Class

```
Public Overridable Function Calculate(num1 As Integer, num2 As Integer) As Integer
```

```
    Form1.TextBox1.Text = ("Virtual method of base class!")
```

```
    Dim result As Integer = 0
```

```
    result = num1 + num2
```

```
    Return result
End Function
```

End Class 'Base Class Ends

১০ম পরিচ্ছেদ: *Object Oriented Programming [OOP]* -বস্তুভিত্তিক প্রোগ্রামিং

বস্তুভিত্তিক প্রোগ্রামিং: *Object Oriented Programming [OOP]*

Public Class Subtract : Inherits Add 'Derived class starts

'override method from Base class

Public Overrides Function Calculate(num1 As Integer, num2 As Integer) As Integer

Form1.TextBox2.Text = ("Override method of Derived class.!")

Dim result As Integer = 0

result = num1 - num2

Return result

End Function

End Class 'Derived class ends

Public Class Form1

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

AddMinus()

End Sub

'called from button clicked event sub

Public Sub AddMinus()

Dim a1 As Add = New Add()

Dim AddRes As Double = (a1.Calculate(400, 100))

TextBox4.Text = "Sum: " & AddRes

Dim a2 As Add = New Subtract() 'instance of the derived class using base class reference

Dim Dif As Double = (a2.Calculate(400, 100))

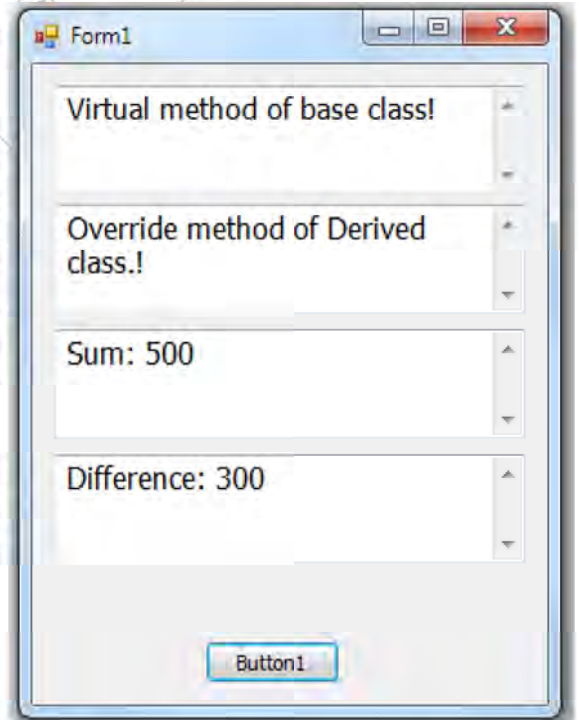
TextBox3.Text = "Difference: " & Dif

End Sub

End Class

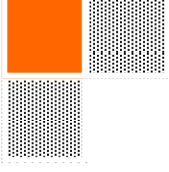
দ্রষ্টব্য: এ পরিচ্ছেদে উপস্থাপিত উদাহরণগুলো মূলত অবজেক্ট অরিয়েন্টেড প্রোগ্রামিংয়ের স্বরূপ বুঝানোর জন্য দেওয়া হয়েছে।

আসলে যেসব ফলাফল এগুলো থেকে পাওয়া যাচ্ছে তা অন্যভাবে অনেকটা সজহ উপায়ে প্রোগ্রাম করার মাধ্যমে অর্জন সম্ভব। আর বাস্তব ক্ষেত্রে এরূপ সাধারণ প্রোগ্রামের জন্য ব্যবহার মোটেই প্রয়োজনীয়ও নয়। যাক, আশাকরি পাঠকরা OOP-এর সাথে সম্পৃক্ত তিনটি মৌলিক বিষয় তথা, **Encapsulation**, **Inheritance** এবং **Polymorphism** সম্পর্কে অত্যন্ত হলেও ধারণা পেয়ে গেছেন। আর এটুকু অর্জনই এ অধ্যায় রচনার উদ্দেশ্য ছিলো। আমরা OOP-নিয়ে খুব একটা মাথা ঘামাবো না। পরবর্তীতে যেসব প্রোগ্রাম উদাহরণ উপস্থাপন করবো তা হবে ভিজুয়াল বেসিক.নেট প্রোগ্রাম করার কৌশল বুঝা, কোড জানা এবং অভিজ্ঞতা অর্জন করার নিমিত্তে। আমরা কিন্তু OOP-ই করবো কিন্তু তা হবে আমাদের অজান্তে!



উপরের প্রোগ্রাম পরিচালনার ফলাফল

১১তম পরিচ্ছেদ: *Working with Files*



ফাইল নিয়ে কাজ : *Working with Files*

ভিজুয়াল বেসিক.নেট দ্বারা ফাইল নিয়ন্ত্রণ: VB.Net File Handling

প্রোগ্রামিং এর একটি গুরুত্বপূর্ণ অংশ হলো তথ্য বা ডাটা ফাইলে লেখা, সংরক্ষণ ও প্রয়োজনে পাঠ করা। যে উপায়ে তা করতে হয় তাকে বলে File Streaming। এর অর্থ, কম্পিউটারের মেমোরিতে সংরক্ষিত ফাইল পড়া [Read] ও লেখা [Write]। প্রত্যেক প্রোগ্রামিং ভাষায় ফাইল নিয়ে কাজ করার কোড আছে। ভিজুয়াল বেসিক.নেটেও ফাইল নিয়ন্ত্রণের জন্য শক্তিশালী স্ট্রিমিং সিস্টেম তৈরি করা হয়েছে। এ পরিচ্ছেদে বিভিন্ন উদাহরণসহ ফাইল নিয়ে কিভাবে কাজ করতে হয় তার ওপর আলোচনা করবো।

File- সংজ্ঞা: নামকরণকৃত মেমোরি ব্লক বা বাক্স যাতে সংরক্ষিত থাকে তথ্যাদি। কম্পিউটারের ডিস্কে [হার্ড ডিস্ক, সিডি, ভিসিডি, পেন-ড্রাইভ] সকল ফাইল প্যাথ [path] দ্বারা সনাক্ত করা হয়।

যেমন: C:\Users\htci\Desktop\Copied\Islamic History Project

Streaming- সংজ্ঞা: প্যাথ [রাস্তা] ব্যবহার করে কোডের মাধ্যমে ফাইল খোলা, লেখা ও পাঠ করাকে বলে স্ট্রিমিং। খোলা ফাইলকে প্রোগ্রামিং ভাষায় স্ট্রিম [Stream] বলে।

ইনপুট/আউটপুট [Input/Output -I/O] ক্লাস

ভিজুয়াল বেসিক.নেটে ফাইল নিয়ন্ত্রণের জন্য আই/অ সুযোগ-সুবিধার মূলে আছে সিস্টেমের নামকৃত [namespace] ক্লাস যাকে System.IO বলে। অর্থাৎ System.IO থেকে সব আই/অ ক্লাস উত্তরাধিকারসূত্রে প্রাপ্ত। যেসব ক্লাস আমরা ব্যবহার করতে পারবো তার মধ্যে উল্লেখযোগ্য ক'টি হলো:

System.IO.Path, System.IO.StreamReader, System.IO.StreamWriter, System.IO.StringReader এবং System.IO.StringWriter। এ পরিচ্ছেদে বিভিন্ন উদাহরণে এগুলোর ব্যবহার ও ব্যাখ্যা আসবে।

ক. স্ট্রিমরাইটার ক্লাস [StreamWriter Class]: এই ক্লাস দ্বারা খোলা ফাইলে [অর্থাৎ স্ট্রিমে] ক্যারাক্টার লেখা হয়। নিচের উদাহরণ দ্বারা বিষয়টি বুঝে আসবে।

Public Class Form1

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    WriteToFile() 'sub call
End Sub
'called from button clicked event sub
Public Sub WriteToFile()
    'creating instance of StreamWriter class, and opening specific file
    Dim FOut As System.IO.StreamWriter = New System.IO.StreamWriter("C:\Users\htci\Desktop\Data.txt")
    FOut.WriteLine("বিসমিল্লাহির রাহমানির রাহীম" & vbCrLf & "ভিজুয়াল বেসিক.নেট একটি সহজ প্রোগ্রামিং ভাষা, অথচ
খুব শক্তিশালী। আপনি অতি সহজে এ ভাষা শিখে নিতে পারেন। প্রয়োজন শুধু শেখার স্পৃহা ও ধৈর্য, এই যা!") 'writing a string to
file
    FOut.Close() 'closing file -don't forget that!
    TextBox3.Text = "হয়তো ফাইল লিখা হয়ে গেছে! ফাইল খুলে চেক করুন। ফাইলের পাথ হলো: " & vbCrLf &
"C:\Users\htci\Desktop\Data.txt"
End Sub
End Class
```


১১তম পরিচ্ছেদ: Working with Files

ফাইল নিয়ে কাজ : Working with Files

Using the OpenFileDialog / SaveFileDialog Class: ফাইল খোলার ডায়ালগের ব্যবহার ফাইল নিয়ন্ত্রণ সম্পর্কে এ পর্যন্ত আমরা যা জেনেছি তাহলো স্ট্রিমরিডার ক্লাসের মাধ্যমে ফাইল খোলা, পড়া ও প্রদর্শন। আমরা এটাও জেনেছি কিভাবে ফাইলে লিখা যায়। এজন্য প্রয়োজন স্ট্রিমরাইটার ক্লাস। তবে উভয় ক্ষেত্রেই আমাদেরকে ফাইলের প্যাথ [path] পুরোটা লিখে দিতে হয়েছে। কিন্তু আমরা সচরাচর দেখে আসছি ওয়ার্ড প্রসেসর, পেইন্ট প্রোগ্রাম ইত্যাদি থেকে ফাইল খুলতে হলে একটি ডায়ালগ আসে। আমরা যে ধরনের ফাইল খুলবো তা চয়েজ করে 'OK' কমান্ডে ক্লিক করলেই ফাইলটি খুলে টেক্সটবক্সে বা ক্যানভাসে প্রদর্শিত হয়। অনুরূপ নতুন ফাইল সংরক্ষণকালেও ফাইল সংরক্ষণের ডায়ালগ আসে। এ দু'টো কোডের মাধ্যমে কিভাবে নিয়ন্ত্রণ করা হয়? এখনই আমরা জেনে নেবো।

ভিজুয়াল বেসিক.নেটে ফাইল খোলা ও সংরক্ষণের দু'টি ক্লাস আছে। এদের নাম যথাক্রমে: **OpenFileDialog()** ও **SaveFileDialog()**। উদাহরণের মাধ্যমে এবার এ দু'টোর ব্যবহার উপস্থাপন করবো।

প্রথমে ফর্মের কোডটি দেখুন:

Public Class Form1

```

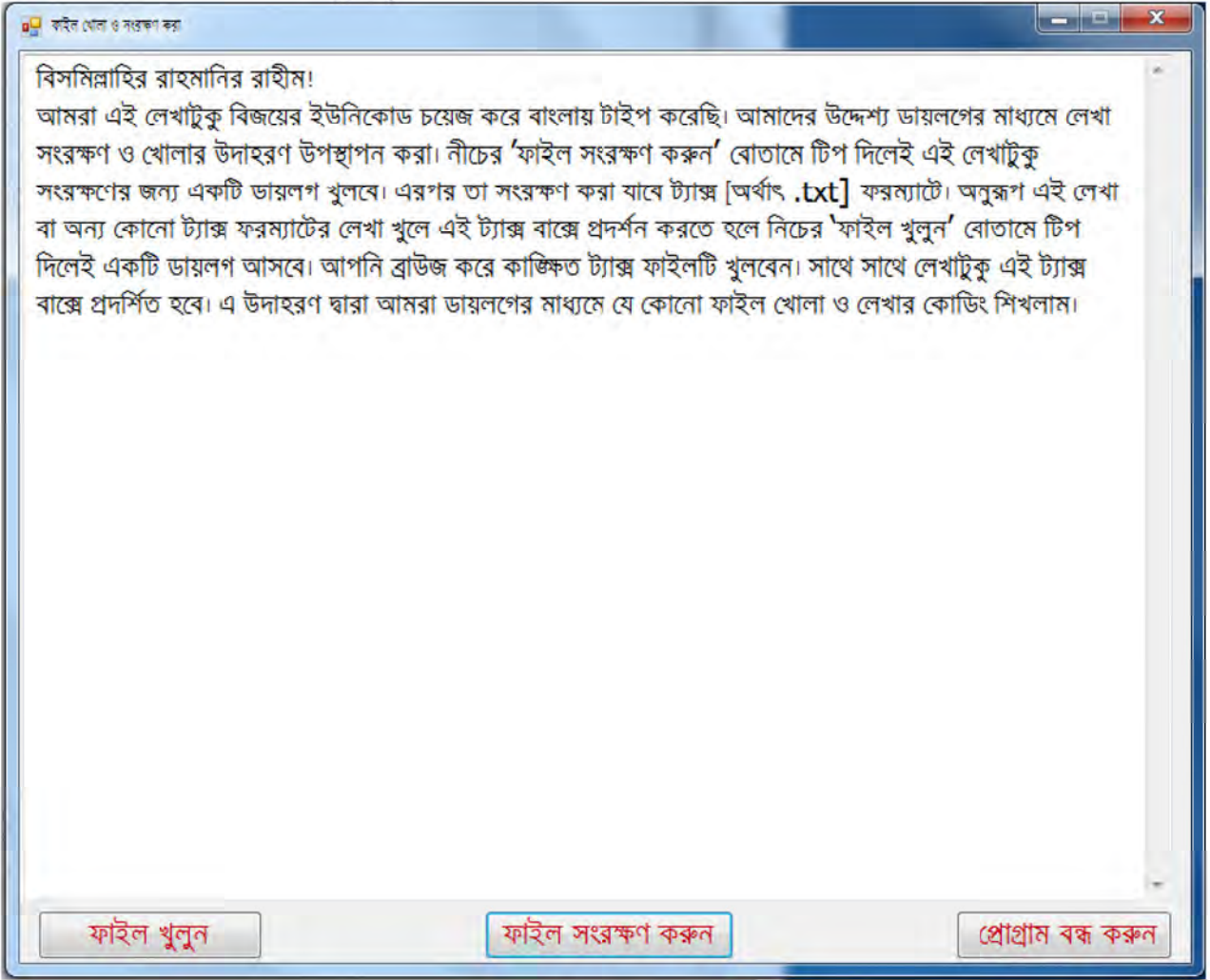
this sub calls the OpenFileDialog Sub which in turn Opens the OpenFileDialog
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    OpenTheFile() 'sub call
End Sub
Sub OpenTheFile() 'this sub displays open dialog
    Dim Open As New OpenFileDialog() 'new instance of file opening dialog class
    Dim myStreamReader As System.IO.StreamReader 'dimension a StreamReader Object
    Open.Filter = "Text[*.*txt]*.txt|All Files [*,*]*.*" 'text files filters
    Open.CheckFileExists = True 'this function makes sure the file exists before opening
    Open.Title = "ফাইল খুলুন" 'Open dialog title text
    Open.ShowDialog(Me) 'shows the dialog
    Try 'Try is a debug keyword. If any problem it will catch it and jump to catch ex as exception line below
        Open.OpenFile() 'open chosen file
        myStreamReader = System.IO.File.OpenText(Open.FileName) 'assign file to StreamReader object
        TextBox1.Text = myStreamReader.ReadToEnd() 'read and display the whole file
    Catch ex As Exception 'any exception will be displayed
    End Try
End Sub
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    If TextBox1.Text <> "" Then 'make sure there is text to be saved
        SaveTheFile() 'sub call
    End If
End Sub
Sub SaveTheFile() 'this sub displays Save Dialog
    Dim Save As New SaveFileDialog() 'new instance of file saving dialog class
    Dim myStreamWriter As System.IO.StreamWriter 'dimension a StreamWriter Object
    Save.Filter = "Text (*.txt)|*.txt|All files (*.*)|*.*" 'text filters
    Save.CheckPathExists = True
    Save.Title = "ফাইল সংরক্ষণ করুন" 'saveDialog title text
    Save.ShowDialog(Me) 'Me means ShowDialog class itself
    Try
        myStreamWriter = System.IO.File.AppendText(Save.FileName) 'append already saved file- shows a replace
    dialog
        myStreamWriter.Write(TextBox1.Text) 'write all text of the textBox to the file
        myStreamWriter.Flush() 'clear memory
    Catch ex As Exception
    End Try
End Sub
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
    Close() 'close form and exit application
End Sub
End Class

```

১১তম পরিচ্ছেদ: Working with Files

ফাইল নিয়ে কাজ : Working with Files

পূর্বের পৃষ্ঠার কোড রানের ফলাফল নিচে দেখানো হয়েছে। আপনি চাইলে একটি নতুন প্রজেক্ট তৈরি করতে পারেন। লক্ষ করুন, ইউনিকোড দিয়ে কিভাবে বাংলা লিখেছি। এছাড়া ইউভোজের বোতামগুলোর টেক্সট প্রোপার্টিও ইউনিকোডের মাধ্যমে বাংলায় করেছি। নিজের ভাষায় এভাবে ইউজার ইন্টারফেস তৈরি করতে পেরে অনেকটা স্বস্তি ও আনন্দবোধ করছি। মাত্র কয়েক বছর পূর্বেও এভাবে করা কঠিন ছিলো। ডিজাইনার ফন্ট [যা মূলত ইংরেজি ক্যারাক্টার] দিয়ে অতীতে কাজ করতে হয়েছে। এখন সব ইউনিকোড প্রায় সকল উইভোজ সিস্টেম সাপোর্ট করে। সুতরাং বাংলায় সফটওয়্যার তৈরি অনেকটা সহজ হয়েছে। আশা করছি ভবিষ্যতে বাংলা ভাষাভাষি প্রোগ্রামাররা এই সুযোগ ব্যবহার করবেন ও সবার জন্য তৈরি করবেন মনোমুগ্ধকর মাতৃভাষা-ভিত্তিক কম্পিউটার সফটওয়্যার।



চিত্র ১৯: আগের পৃষ্ঠার কোড রান করার ফলাফল। লক্ষ করুন: জানালার পুরোটাই বাংলায় উপস্থাপিত হয়েছে। এজন্য আমরা বাংলা ফন্ট চয়েজ করি নি। শুধুমাত্র বিজয়ের ইউনিকোড টাইপ সিস্টেম অন করেছি। মাইক্রোসফটের ভিজুয়াল স্টুডিও এক্সপ্রেস ফর ডেস্কটপ ইউনিকোড সাপোর্ট করে। এ কারণেই বাংলায় সবকিছু সম্ভব।

ভিজুয়াল বেসিক.নেটে ফাইল ম্যানেজমেন্ট সম্পর্কে আরো একদুটো ব্যাপার জানার আছে। পরবর্তী পৃষ্ঠাসমূহে উদাহরণসহ এগুলো তুলে ধরছি।

১১তম পরিচ্ছেদ: *Working with Files*

ফাইল নিয়ে কাজ : *Working with Files*

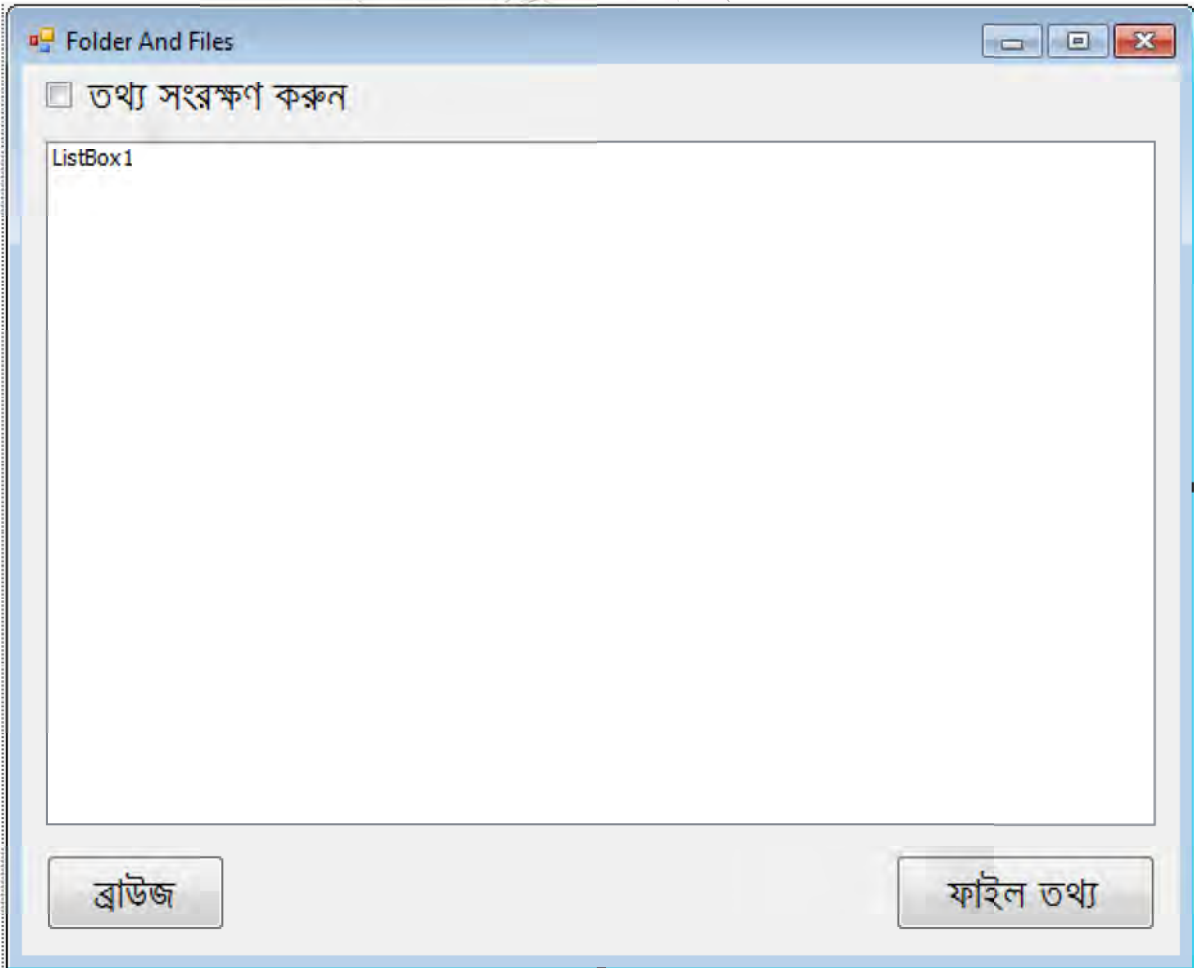
উদাহরণ-১: `My.Computer.FileSystem` Object ব্যবহার করে ফাইল নিয়ন্ত্রণ

আমরা ইতোমধ্যে ফাইল নিয়ন্ত্রণ সম্পর্কিত যেসব ব্যাখ্যা ও উদাহরণ তুলে ধরেছি তা সবই ছিলো **System.IO namespace** -এর অন্তর্ভুক্ত। ডিজুয়াল বেসিক.নেটে ফাইল নিয়ে কাজ করার আরো একটি পদ্ধতি আছে। একে বলে **My.Computer.FileSystem Object**। নিম্নে এ সম্পর্কিত একটি উদাহরণ তুলে ধরছি। এসাথে শেষের দিকে একই কাজ সারতে প্রথম পদ্ধতি তথা **System.IO** -এর অন্তর্ভুক্ত কোডও দেওয়া হয়েছে। শিক্ষার্থীরা উভয় কোডের মধ্যে পার্থক্য লক্ষ্য করবেন।

এবার আমরা ধারাবাহিক পদক্ষেপে পুরো উদাহরণটি লিপিবদ্ধ করবো। শিক্ষার্থীরা নিচের প্রতিটি পদক্ষেপ সঠিকভাবে অনুসরণ করে এ প্রজেক্টটি তৈরি করুন।

ক. একটি নতুন উইন্ডোজ ফর্ম প্রজেক্ট তৈরি করুন। প্রজেক্টের নামকরণ করুন 'FolderAndFiles'। অবশ্য চাইলে অন্য নামও দিতে পারেন। প্রদর্শিত ফর্মের প্রোপার্টি হবে নিম্নরূপ:

খ. ফর্মে টুল বাক্স থেকে সংযুক্ত করুন একটি `ListBox`, একটি `CheckBox`, দু'টি `Button` এবং একটি `FolderBrowserDialog` কন্ট্রোল। টুল বাক্সে যেয়ে এগুলোর নামের ওপর ডবল ক্লিক করলেই ফর্মে এসে যুক্ত হবে। সকল কন্ট্রোলের নাম নিচের জানালার অনুরূপ করুন। আমি বাংলা নামকরণ করেছি। চাইলে আপনি অন্য নাম বা এমনকি ইংরেজিতেও নামকরণ করতে পারেন।



১১তম পরিচ্ছেদ: *Working with Files*

ফাইল নিয়ে কাজ : *Working with Files*

```

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    If ListBox1.SelectedItem Is Nothing Then 'make sure user has selected an item
        MessageBox.Show("Please select a file.")
    Exit Sub
End If
Dim filePath = ListBox1.SelectedItem.ToString ' Obtain the file path from the list box selection.
' Verify that the file was not removed since the " button was clicked.
If My.Computer.FileSystem.FileExists(filePath) = False Then
    MessageBox.Show("File Not Found: " & filePath)
    Exit Sub
End If
Dim fileInfoText As String = GetTextForOutput(filePath) ' Obtain file information in a string.
MessageBox.Show(fileInfoText) ' Show the file information.
'The if-end if block of code sets the log file path to put it in the same directory as that of the selected file.
'The text of the log entry is set to the current date and time followed by the file information.
'The WriteAllText method, with the append argument set to True, is used to create the log entry.
If CheckBox1.Checked = True Then
    ' Place the log file in the same folder as the examined file.
    Dim logFolder As String = My.Computer.FileSystem.GetFileInfo(filePath).DirectoryName
    Dim logFilePath = My.Computer.FileSystem.CombinePath(logFolder, "log.txt")
    Dim logText As String = "Logged: " & Date.Now.ToString & vbCrLf & fileInfoText & vbCrLf & vbCrLf
    ' Append text to the log file.
    My.Computer.FileSystem.WriteAllText(logFilePath, logText, append:=True)
End If
End Sub
Private Function GetTextForOutput(ByVal filePath As String) As String
    If My.Computer.FileSystem.FileExists(filePath) = False Then
        Throw New Exception("File Not Found: " & filePath) ' Verify that the file exists. In no, show dialog & exit
    End If
    ' Create a new StringBuilder, which is used to efficiently build strings.
    Dim sb As New System.Text.StringBuilder()
    ' Obtain file information.
    Dim thisFile As System.IO.FileInfo = My.Computer.FileSystem.GetFileInfo(filePath)
    ' Add file attributes.
    sb.Append("File: " & thisFile.FullName)
    sb.Append(vbCrLf)
    sb.Append("Modified: " & thisFile.LastWriteTime.ToString)
    sb.Append(vbCrLf)
    sb.Append("Size: " & thisFile.Length.ToString & " bytes")
    sb.Append(vbCrLf)
    ' Open the text file.
    Dim sr As System.IO.StreamReader =
        My.Computer.FileSystem.OpenTextFileReader(filePath)
    If sr.Peek() >= 0 Then
        sb.Append("First Line: " & sr.ReadLine()) ' Add the first line from the file.
    End If
    sr.Close()
    Return sb.ToString
End Function

```

১১তম পরিচ্ছেদ: Working with Files

ফাইল নিয়ে কাজ : Working with Files

```
'enable or disable Button1 and CheckBox wether ListBox1 item selected or not respectively
Private Sub SetEnabled()
    'AnySelected becomes Yes or No depending wether any Item selected or not reselectively
    Dim AnySelected As Boolean = (ListBox1.SelectedItem IsNot Nothing)
    Button1.Enabled = AnySelected 'if AnySelected True, Button1 is Enabled
    CheckBox1.Enabled = AnySelected 'if AnySelected True, CheckBox1 is Enabled too
End Sub
Private Sub ListBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
ListBox1.SelectedIndexChanged
    SetEnabled() 'CheckBox1 and Button1 enabled
End Sub
End Class
```

কোডের ব্যাখ্যা: ফর্মে লোড ইভেন্ট হ্যান্ডলারে আমরা একটিমাত্র লাইন লিখেছি। লাইনটি এই:

```
FolderBrowserDialog1.SelectedPath = My.Computer.FileSystem.CurrentDirectory
```

এখানে ফন্ডারব্রাউজারডায়লগ ক্লাসের ডিফোল্ট প্যাথ বলে দেওয়া হয়েছে। ফাইল সিস্টেমের .CurrentDirectory যে ডাইরেক্টরী থেকে লিস্ট বাক্সে ফাইলের প্যাথ নিয়ে আসা হয়েছে সেই প্যাথের দিকেই নির্দেশ করে। অন্যকথায় প্রথমবার যে ডাইরেক্টরিতে যেয়ে আপনি ফন্ডার খুলবেন, সে-ই ডাইরেক্টরিকেই দ্বিতীয়বার দেখাবে। আর লগ টেক্সটটিও সেখানে সংরক্ষিত হবে। আমরা যদি My.Computer.FileSystem.CurrentDirectory এর বদলে এখানে System.IO ব্যবহার করি তাহলে লিখবো:

```
FolderBrowserDialog1.SelectedPath = System.IO.Directory.GetCurrentDirectory()
```

GetCurrentDirectory() -সিস্টেম.আইও এর একটি ফাংশন। এটা রিটার্ন করে বর্তমান ডাইরেক্টরির প্যাথ। সাবের দ্বিতীয় লাইনে অপর একটি সাবরুটিন কল করা হয়েছে। এটি 'ফাইল তথ্য' বোতাম ও 'তথ্য সংরক্ষণ করুন' চেক বাক্সকে অকার্যকর করে। এগুলো কার্যকর হবে এই একই সাব পুনরায় কল করে- যখন ব্যবহারকারী কোনো ফাইলের তথ্য জানার জন্য লিস্ট বাক্সের কোনো আইটেমে ক্লিক করবে।

Button2 -এর ক্লিক ইভেন্ট সাবে; If FolderBrowserDialog1.ShowDialog() = DialogResult.OK Then -অংশ প্রথমে ডায়লগটি দেখাবে। এরপর ব্যবহারকারী কোনো ফন্ডার চয়েজ করে 'OK' বোতামে ক্লিক করলে বুলিয়ান ফলাফল 'True' হবে। সুতরাং পরের লাইনে যেয়ে ListFile() সাবটি কল করবে- যার কাজ প্রাপ্ত ফাইলগুলোর নাম লিস্ট বাক্সে প্রদর্শন করা। আর যদি 'Cancel' বোতামে ক্লিক করা হয় তাহলে কিছুই হবে না। শুধুমাত্র ডায়লগটি বন্ধ হবে। একই সময় SetEnabled() সাব কল করে নিশ্চিত করা হবে 'ফাইল তথ্য' বোতাম ও তথ্য 'সংরক্ষণ করুন' চেক বাক্স যাতে Disabled অবস্থায় থাকে। অবশ্য ডায়লগের ফলাফল 'OK' কিংবা 'Cancel' হোক উভয় অবস্থায়ই Button2-তে ক্লিক করলে এ দু'টো Disabled থাকা জরুরী- আর তা-ই হবে। কারণ আমরা কল করছি End If কথাটির পরে।

ListFiles(ByVal folderPath As String) সাবের কাজ হলো প্রদত্ত ফন্ডার প্যাথ [সাবের আর্গুমেন্ট] ব্যবহার করে ঐ ফন্ডানের .txt, .Doc & .Docx ফাইল টাইপের যতো ফাইল আছে সবগুলোর প্যাথ সংগ্রহ করে লিস্ট বাক্সে প্রদর্শন করা। এজন্য অপারেটিং সিস্টেমের My.Computer.FileSystem.GetFiles ফাংশনটি কাজে লাগানো হয়েছে। এ ফাংশনের গুরুত্বপূর্ণ দু'টি আর্গুমেন্ট হলো: ফন্ডারের প্যাথ [যা আসবে FolderBrowserDialog1.SelectedPath থেকে] ও অনুসন্ধান ফিল্টার তথ্য [এখানে] .txt, .Doc & .Docx ফাইল টাইপ। অপর আর্গুমেন্ট নিশ্চিত করে শুধুমাত্র উচ্চ পর্যায়ের ডাইরেক্টরী থেকে তথ্য আনা হবে। অন্যকথায় অনুসন্ধান হবে শুধুমাত্র কারেন্ট ফন্ডারে- এর ভেতরস্থ অপর কোনো ফন্ডার এ অনুসন্ধানের আওতাভুক্ত থাকবে না।

১২তম পরিচ্ছেদ: *Accessing Computer Resources*



কম্পিউটারের রিসোর্স সংগ্রহ ও ব্যবহার : *Accessing Computer Resources*

My.Computer.Object

সিস্টেমের এই বস্তুটি আমাদেরকে সুযোগ দিয়েছে কম্পিউটারের বিভিন্ন প্রোপার্টিকে ব্যবহার ও নিয়ন্ত্রণ করার। যেমন: অডিও, ক্লিপবোর্ড, ঘড়ি, ফাইল সিস্টেম, তথ্য, কিবোর্ড, মাউস, নেটওয়ার্ক, পোর্ট, রেজিস্ট্রি ইত্যাদি। এসব নিয়ন্ত্রণের কিছু উদাহরণ এবার উপস্থাপন করছি।

১. My.Computer.Audio: নিশ্চয়ই আপনি লক্ষ্য করেছেন, কোনো কোনো প্রোগ্রামের মধ্যে অডিও প্লে করতে। ভিজুয়াল বেসিক.নেটে এটা যে কোড দ্বারা করা হয় তা-ই হলো এই বস্তুটি। নিচের উদাহরণ দ্বারা ব্যবহার পদ্ধতি আমার শিখে নেবো।

মনে করুন আমরা আল-কুরআনের একটি সুরা শুনবো। সুরাটির অডিও ফাইল সংরক্ষিত আছে যেখানে তার প্যাথ হলো: C:\Users\htci\Music\Muslima - Nas.mp3। আমরা এই ফাইলটি প্লে করতে পারি এই কোডটি দ্বারা:

Sub PlayBackgroundSoundFile()

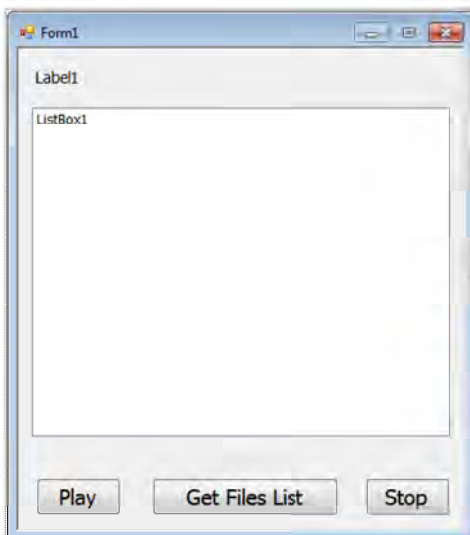
```
My.Computer.Audio.Play("C:\Users\htci\Music\Muslima - Nas.mp3",
AudioPlayMode.WaitToComplete) 'শেষের এই প্র্যারামিটার নিশ্চিত করে ফাইল প্লে শেষ না হওয়া পর্যন্ত
'পরবর্তী কোড রান করবে না।
```

End Sub

উপরোক্ত কোড ব্যবহার করে আমরা তৈরি করতে পারি একটি সাধারণ অডিও প্লেয়ার প্রোগ্রাম। আসুন তাহলে, এটা করে নিই।

উদাহরণ-১: Simple .wav audio player -.wav ফাইল অডিও প্লেয়ার

আপনার সকল অডিও ফাইলগুলো যে কোনো একটি ফন্ডারে সংরক্ষণ করে রাখতে পারেন। মনে রাখবেন ফাইলগুলো হতে হবে .wav ফাইল। আপনার অডিও ফাইল এই ফরমেটে না থাকলে ইন্টারনেট থেকে ফ্লিওয়্যার অডিও কনভার্টার ফাইল ডাউনলোড করতে পারেন। বেশ কয়েকটি ফ্লিওয়্যার আছে। আমি ব্যবহার করি- 'frea:ac - free audio converter v 1.0.16'। যাক, নিচের পদক্ষেপগুলো একে একে অনুসরণ করে প্রজেক্ট তৈরি করুন।



১. নতুন একটি উইন্ডোজ ফর্ম প্রজেক্ট তৈরি করুন- এর নামকরণ করুন: wavAudioPlayer।
২. ফর্মটির ডিজাইন পাশের ছবিতে প্রদর্শিত হয়েছে। এতে থাকবে একটি লেবেল, একটি লিস্ট বাক্স ও ৩টি বোতাম।
৩. Button1 ডবল ক্লিক করুন। ফর্মের ক্লাসের ভেতর বোতামের ক্লিক ইভেন্ট সাব আসবে। এতে লিখুন নিম্নের কোডটুকু।

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    If ListBox1.SelectedItem = "" Then Exit Sub 'make sure a file path is
chosen
    PlayFile() 'call sound play sub
End Sub
```

১২তম পরিচ্ছেদ: *Accessing Computer Resources*

কম্পিউটারের রিসোর্স সংগ্রহ ও ব্যবহার : *Accessing Computer Resources*

প্রথম লাইন দ্বারা নিশ্চিত করা হয়েছে যে, ব্যবহারকারী লিস্ট বাক্সের কোনো একটি লাইনে ক্লিক করে সিলেক্ট করেছেন। এটা নিশ্চিত হওয়ার পর পরের লাইনে আমরা নিম্নের সাবটি কল করেছি।

Sub PlayFile()

```
'get the path string from list box selected line
Dim FileP As String = ListBox1.SelectedItem
My.Computer.Audio.Play(FileP) 'play file
Label1.Text = "Now Playing: " & FileName(FileP) 'call FileName() sub show info
End Sub
```

উক্ত সাবের প্রথম লাইনে FileP নামক স্ট্রিং ভেরিয়েবলের মধ্যে লিস্ট বাক্সের ফাইল প্যাথ স্ট্রিংটি সংরক্ষণ করেছি। এরপর সিস্টেমের My.Computer.Audio.Play ফাংশনটি কল করেছি ও এর প্যারামিটার হিসাবে FileP -অর্থাৎ ফাইলের প্যাথ দিয়েছি। সুতরাং .wav ফাইলটি খুলে তা বাজতে শুরু হবে। পরের লাইনটি যে ফাইল প্লে হচ্ছে তার নাম লেবেলে দেখাবে। এবার লিখুন বোতাম২ এর ক্লিক ইভেন্ট সাবের নিচের কোড লাইনটি। এই কোড লাইন বলছে ফাইল প্লে করা বন্ধ করতে।

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click

```
My.Computer.Audio.Stop() 'stops playing
End Sub
```

বোতাম৩ এর ক্লিক ইভেন্টে যে কোডটি দেওয়া হয়েছে তার দ্বারা উদ্দেশ্য হলো, ফন্ডার ব্রাউজার ডায়ালগ খোলা ও যে কোনো ফন্ডারে প্রাপ্ত সকল .wav সাউন্ড ফাইলের প্যাথ স্ট্রিং এনে লিস্ট বাক্সে প্রদর্শন করা। সুতরাং নিচের কোড ব্লকটি বোতাম৩ এর ইভেন্ট হ্যান্ডলারে লিখতে হবে:

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

```
If FolderBrowserDialog1.ShowDialog() = DialogResult.OK Then 'if Cancel, the code will not execute
'List files in the folder.
ListFiles(FolderBrowserDialog1.SelectedPath) 'call the sub below
End If
End Sub
```

প্রথম লাইনের '=' চিহ্নের আগের অংশ ফন্ডার ব্রাউজার ডায়ালগটি খুলবে। এরপর ফলাফল যদি 'OK' হয়- অর্থাৎ ব্যবহারকারী কোনো ফন্ডার খুলেন তাহলে পরবর্তী লাইনের কোড রান করবে- অন্যথায় নয়। আমাদের তৈরি ListFiles() সাবরুটিনটি একটিমাত্র প্যারামিটার গ্রহণ করে- সেটা হলো সংরক্ষিত ফাইলের প্যাথ- যা আসবে ফন্ডার ব্রাউজার থেকে। এই সাবরুটিনটি নিম্নরূপ:

'opens all .wav files in a folder and list their path in the list Box

Private Sub ListFiles(ByVal folderPath As String)

```
ListBox1.Items.Clear() 'clear any previous list
Dim fileNames As String() = System.IO.Directory.GetFiles(folderPath, "*.wav", _
System.IO.SearchOption.TopDirectoryOnly) 'this line gets the path list
For Each fileName As String In fileNames 'this one puts the list line by line in the ListBox
Listbox1.Items.Add(fileName)
Next
End Sub
```

এই সাবরুটিনের প্রথম লাইন আগের কোনো লেখা থাকলে তা লিস্ট বাক্স থেকে মুছে ফেলে। দ্বিতীয় লাইন ফন্ডার ব্রাউজার থেকে সিলেক্ট করা ফন্ডারের ভেতর যতো .wav টাইপের সাউন্ড ফাইল আছে সবগুলোর প্যাথ সংগ্রহ করে। আর For Each -- Next লুপটির কাজ হলো সবগুলো ফাইলের প্যাথ স্ট্রিং, লাইলে লাইনে লিস্টবাক্সে লিখে দেওয়া। সর্বশেষ আমাদের বানানো ফাংশনটির ব্যবহার মূলত জরুরি নয়। তবে আমরা তো শিখবো! প্যাথ স্ট্রিং থেকে শুধু ফাইলের নাম কিভাবে বের করা যায় তা-ই দেখানো হলো এ ফাংশন তৈরির উদ্দেশ্যে।

১২তম পরিচ্ছেদ: *Accessing Computer Resources*

কম্পিউটারের রিসোর্স সংগ্রহ ও ব্যবহার : *Accessing Computer Resources*

প্রথমে ফাংশনটি দেখে নিই। এরপর এর ব্যাখ্যা আসবে।

'this function returns the name of a file from path string parameter

```
Private Function FileName(SupplyString As String) As String
    Dim A As Integer : Dim RetString As String = ""
    For A = Len(SupplyString) To 1 Step -1
        If Mid(SupplyString, A, 1) = "\" Then Exit For 'last \' char position
    Next
    RetString = Mid(SupplyString, (A + 1), Len(SupplyString) - A) 'get the name
    Return RetString
End Function
```

ফাংশনের ব্যাখ্যা: প্রত্যেক প্যাথ নামের শেষের অংশে '\' -এ চিহ্নটি পরে ফাইলের নাম থাকে। সুতরাং আমরা চাই ফাংশনের প্যারামিটার হবে প্যাথ নামটি যার টাইপ হলো স্ট্রিং। ফাংশন থেকে যে ভ্যালু রিটার্ন হবে তা-ও একটি স্ট্রিং টাইপ। এজন্য আমরা প্রথম লাইনে লিখেছি:

Private Function FileName(SupplyString As String) As String

উক্ত লাইনের প্রথম দু'টি শব্দ লিখা বাধ্যতামূলক। এরপর এসেছে ফাংশনের নাম- এখানে FileName। বন্ধনীর ভেতর থাকবে প্যারামিটার- বা আর্গুমেন্ট যা কলিং কোডে দিতে হবে। কী দিতে হবে, তা আমরা সাব্যস্ত করে দিয়েছি- সেটা হলো স্ট্রিং টাইপ। এজন্য আমরা লিখেছি: (SupplyString As String)। এ তিনটি শব্দের প্রথমটি আমাদের বানানো, দ্বিতীয় ও তৃতীয়টি বাধ্যতামূলক- এখানে As String। কারণ, আমরা চাই SupplyString -এর টাইপ হবে স্ট্রিং। লাইনের শেষের দু'টি শব্দ দ্বারা আমরা জানিয়ে দিচ্ছি ফাংশন কোন টাইপের ভ্যালু রিটার্ন করবে- তা-হবে, স্ট্রিং। ফাংশনের দ্বিতীয় লাইনে দু'টি ভেরিয়েবল ঘোষণা করেছি: A - একটি ইন্টজার টাইপ ও RetString -একটি স্ট্রিং টাইপ। শেষোক্তটি শূন্য স্ট্রিং হিসাবে ইনিশিয়েলাইজ করেছি। প্রথমটি পরবর্তী ফর-নেক্সট লুপটির জন্য ব্যবহৃত হয়েছে। এ ফর-নেক্সট লুপের প্রথম লাইন দ্বারা বুঝাচ্ছি, SupplyString তথা প্যাথ নামের দৈর্ঘ্যসংখ্যা থেকে ১ পর্যন্ত এই লুপটি রান করবে। আমাদের উদ্দেশ্য, ঠিক কোন পজিশনে স্ট্রিংয়ের মধ্যে শেষের '\' -এ চিহ্নটি আছে তা নির্ণয় করা। সুতরাং আমরা ডান থেকে শুরু করে একে একে বামদিকে যেয়ে উক্ত চিহ্নটির পজিশন খুঁজে দেখবো। যে মুহূর্তে এটা পাওয়া যাবে তখনই লুপ থেকে বের হবো। আর এ চিহ্নটি বার বার পরীক্ষার জন্য আমরা পরের লাইনের কোডটি ব্যবহার করেছি। তাহলো:

If Mid(SupplyString, A, 1) = "\" Then Exit For -এখানে, Mid ফাংশনের কাজ হলো স্ট্রিংয়ের পজিশন A -এর একটি মাত্র অক্ষর সংগ্রহ করা। আমরা বলছি এই অক্ষর বা ক্যারাক্টারটি যদি '\' হয়, তাহলে লুপ থেকে বের হও। মনে রাখবেন আমাদের ইন্টজার ভেরিয়েবল A শুরু হয়েছে প্যাথ স্ট্রিং এর মোট অক্ষরসংখ্যা থেকে। আর এ সংখ্যার মান প্রত্যেক লুপে এক এক করে কমে আসছে। সুতরাং উক্ত '\' -ক্যারাক্টারটি যেখানে পাওয়া যাবে সেখানকার সংখ্যা-মান হবে A। অর্থাৎ ঠিক কোন সর্বডানের পজিশনে '\' -এ চিহ্নটি আছে তার মান আমরা পেলাম। এই মানটি বের করাই হলো ফাংশনের মূল কাজ।

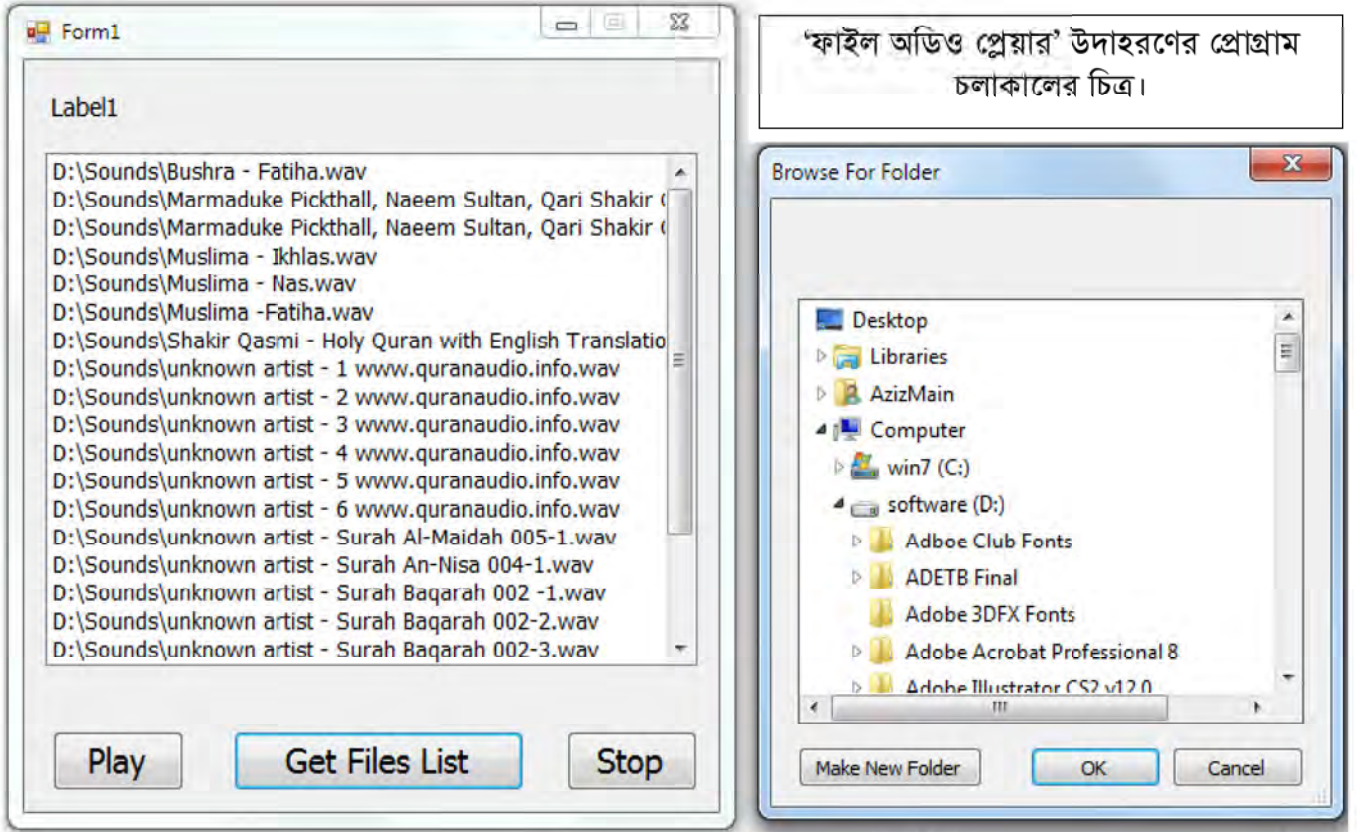
এবার আমরা ফাইলের নাম অংশ একটিমাত্র কোড লাইনে সংগ্রহ করতে পারি:

RetString = Mid(SupplyString, (A + 1), Len(SupplyString) - A) -এখানে Mid ফাংশনের কাজ হলো পুরো স্ট্রিং থেকে প্রাপ্ত সংখ্যা (A+1)-এ শুরু করে স্ট্রিং এর শেষ পর্যন্ত [Len(SupplyString) - A] সংগ্রহ করা। আমরা একে [অর্থাৎ ফাইলের নামকে] সংরক্ষণ করেছি RetString ভেরিয়েবলে। পরের লাইনে ফাংশন তার রিটার্ন ভ্যালু ফিরিয়ে দেবে। তাহলো এই সংরক্ষিত RetString ভেরিয়েবল: **Return RetString**।

যাঁক, আশাকরি ফাংশনের ওপর এই বিস্তারিত ব্যাখ্যা আপনার জন্য ফাংশন কিভাবে কাজ করে তা বুঝতে সহায়ক হবে। পরের পৃষ্ঠায় একটি চিত্রে এ উদাহরণ প্রোগ্রামটির রান-টাইমের ছবি তুলে ধরেছি।

১২তম পরিচ্ছেদ: *Accessing Computer Resources*

কম্পিউটারের রিসোর্স সংগ্রহ ও ব্যবহার : *Accessing Computer Resources*



উদাহরণ-২: *Me.Computer* -এর বিভিন্ন ফাংশন, মেথড ও প্রোপার্টি

Me.Computer-এর বিভিন্ন উদাহরণ একত্রে একটি প্রজেক্টে তুলে ধরেছি। পাঠকদের উচিত একটি উইন্ডোজ ফর্ম প্রজেক্ট সৃষ্টি করে নিচের পদক্ষেপ মুতাবিক এটি ধীরে ধীরে ডেভেলপ করা।

- (ক) `My.Computer.Info.AvailablePhysicalMemory` এটা একটি রিড-অনলি [শুধু পাঠ্য তথ্য] প্রোপার্টি। কম্পিউটারে কি পরিমাণ বাস্তব মুক্ত মেমোরি আছে তা এই প্রোপার্টির মাধ্যমে জানা যায়।
- (খ) `My.Computer.Info.TotalPhysicalMemory` এটা মোট বাস্তব মেমোরি কত রিটার্ন করে।
- (গ) `My.Computer.Info.AvailableVirtualMemory` এটা রিটার্ন করে মুক্ত ভারচুয়াল মেমোরির মাত্রা।
- (ঘ) `My.Computer.Info.TotalVirtualMemory` এটা থেকে জানা যায় মোট ভারচুয়াল মেমোরি কত।

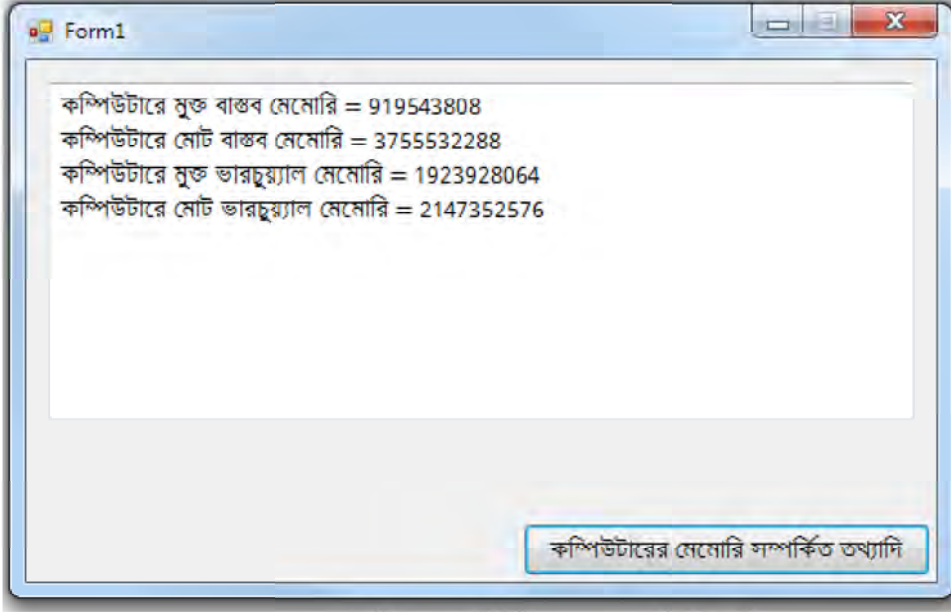
উপরোক্ত প্রোপার্টিগুলো কিভাবে কাজ করে দেখার জন্য প্রথমে প্রজেক্টের জানালায় একটি বোতাম ও একটি টেক্সট বাক্স যুক্ত করুন। এরপর বোতামের ক্লিক ইভেন্টে [বোতামে ডবল ক্লিক করলেই কোড জানালায় ক্লিক ইভেন্ট সাব আসবে] নিচের কোড ব্লকটি লিখুন।

```

'information on the memories of user's computer in bytes.
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    TextBox1.Text = "" 'clear any text
    TextBox1.Text += ("কম্পিউটারে মুক্ত বাস্তব মেমোরি = " & _
    My.Computer.Info.AvailablePhysicalMemory) & vbCrLf
    TextBox1.Text += ("কম্পিউটারে মোট বাস্তব মেমোরি = " & _
    My.Computer.Info.TotalPhysicalMemory) & vbCrLf
    TextBox1.Text += ("কম্পিউটারে মুক্ত ভারচুয়াল মেমোরি = " & _
    My.Computer.Info.AvailableVirtualMemory) & vbCrLf
    TextBox1.Text += ("কম্পিউটারে মোট ভারচুয়াল মেমোরি = " & _
    My.Computer.Info.TotalVirtualMemory)
End Sub
    
```

১২তম পরিচ্ছেদ: *Accessing Computer Resources*

কম্পিউটারের রিসোর্স সংগ্রহ ও ব্যবহার : *Accessing Computer Resources*



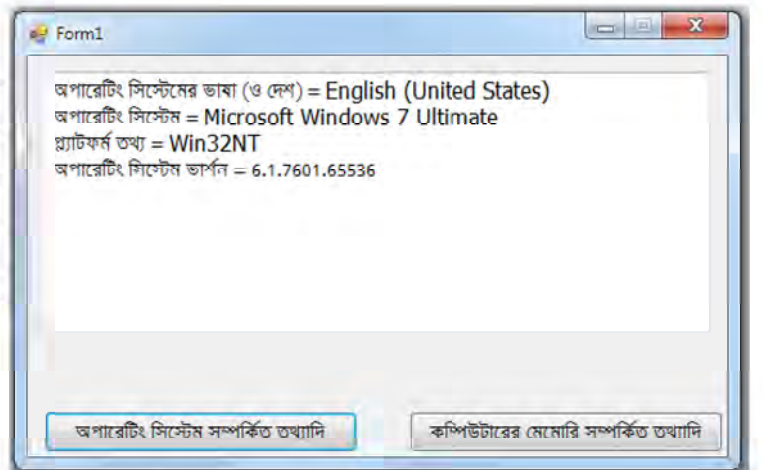
আগের পৃষ্ঠার কোড ব্লক 'কম্পিউটারের মেমোরি সম্পর্কিত তথ্যাদি' বোতামের ক্লিক ইভেন্টে ক্লিক করে রান করার ফলাফল।

- (ঙ) `My.Computer.Info.InstalledUICulture.DisplayName` এটা দ্বারা কম্পিউটারের অপারেটিং সিস্টেমের ভাষা ও কোন্ দেশের ভাষা জানা যায়।
- (চ) `My.Computer.Info.OSFullName` এটা অপারেটিং সিস্টেমের পূর্ণাঙ্গ নাম রিটার্ন করে।
- (ছ) `My.Computer.Info.OSPlatform` এটা দ্বারা অপারেটিং সিস্টেমের প্ল্যাটফর্ম কি তা জানা যায়।
- (জ) `My.Computer.Info.OSVersion` এটা দিয়ে আমরা জানতে পারি অপারেটিং সিস্টেমের ভার্সন কোন্টি।

উপরোক্ত প্রতিটি তথ্য জানতে ফর্মে আরেকটি বোতাম যুক্ত কর। এর ক্লিক ইভেন্টে লিখুন:

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    TextBox1.Text = "" 'clear any text
    TextBox1.Text += ("অপারেটিং সিস্টেমের ভাষা (ও দেশ) = " & _
    My.Computer.Info.InstalledUICulture.DisplayName) & vbCrLf
    TextBox1.Text += ("অপারেটিং সিস্টেম = " & _
    My.Computer.Info.OSFullName) & vbCrLf
    TextBox1.Text += ("প্ল্যাটফর্ম তথ্য = " & _
    My.Computer.Info.OSPlatform) & vbCrLf
    TextBox1.Text += ("অপারেটিং সিস্টেম ভার্সন = " & _
    My.Computer.Info.OSVersion)
End Sub
```

লক্ষ্যণীয় যে, একই লাইনে দীর্ঘ কোড না লিখে আমরা (_) আন্ডারস্কোর দ্বারা লাইনটি ভাগ করে একাধিক লাইনে লিখতে পারি। উপরের কোডে চারটি লাইন এভাবে ভাগ করে জোড়া জোড়া লাইনে লিখেছি। আন্ডারস্কোর থাকতে হবে লাইনের শেষে। যাক, উপরের কোড রান করার ফলাফল ডানের চিত্রে তুলে ধরেছি। একটু দেখে নিন।



১২তম পরিচ্ছেদ: *Accessing Computer Resources*

কম্পিউটারের রিসোর্স সংগ্রহ ও ব্যবহার : *Accessing Computer Resources*

(২) **My.Computer.Keyboard.CapsLock** -ক্যাপ্স লক প্রেস করা আছে কি না সে সম্পর্কে 'সত্য' বা 'মিথ্যা' বুলিয়ান ফলাফল দেবে। যেমন:

```
If My.Computer.Keyboard.CapsLock Then MsgBox("Caps Lock is ON")
```

(৩) **My.Computer.Keyboard.CtrlKeyDown** -কন্ট্রোল কি প্রেস করা আছে কি না সে সম্পর্কে 'সত্য' বা 'মিথ্যা' বুলিয়ান ফলাফল দেবে। যেমন:

```
If My.Computer.Keyboard.CtrlKeyDown Then MsgBox("Control key is down")
```

এভাবে আরো আছে, যাদের ফলাফল হচ্ছে বর্ণিত কি-টি চাপানো আছে কি না:

(৪) **My.Computer.Keyboard.NumLock**

(৫) **My.Computer.Keyboard.CtrlScrollLock**

(৬) এবং **My.Computer.Keyboard.ShiftKeyDown**

(৪) **My.Computer.Mouse** -এটির দ্বারা কম্পিউটারের মাউস সম্পর্কে জানা ও সেট করার দু'টি প্রোপার্টি জানা যায়। যেমন:

(১) **My.Computer.Mouse.WheelExists** -মাউসের মধ্যে স্ক্রল চাকা আছে কি না? থাকলে 'True' না থাকলে 'False'।

(২) **My.Computer.Mouse.WheelScrollLines** -মাউসের স্ক্রল চাকা প্রতি চক্রে কতটুকু ঘুরবে সে তথ্য দেবে। উদাহরণ হিসেবে নিচের কোড ব্লক ব্যবহার করে দেখুন। একটি বোতামের ইভেন্ট কোডে এটি লিখে রান করুন।

```
If My.Computer.Mouse.WheelExists Then
    Dim lines As Integer = My.Computer.Mouse.WheelScrollLines
    If lines > 0 Then
        MsgBox("Application scrolls " &
            lines & " line(s) for each wheel turn.")
    Else
        MsgBox("Application scrolls " &
            (-lines) & " page(s) for each wheel turn.")
    End If
Else
    MsgBox("Mouse has no scroll wheel.")
End If
```

(৪) **My.Computer.Network** - যে কম্পিউটারে কোড রান করবে তাতে নেটওয়ার্ক আছে কি না, থাকলে নির্দিষ্ট ইউআরএল ঠিকানা থেকে ফাইল ডাউনলোড করা ও ফাইল আপলোড করা ইত্যাদি কাজ আঞ্জাম দেওয়ার সুযোগ করে দিয়েছে এই মেথডটি। নিচে আমরা তিনটি ফাংশন উদাহরণসহ তুলে ধরেছি।

(১) **My.Computer.Network.IsAvailable** - এটা দ্বারা আমরা জানতে পারি বর্তমান সময়ে নেটওয়ার্ক আছে কি না? যেমন:

```
If My.Computer.Network.IsAvailable = True Then
    MsgBox("Computer is connected.")
Else
    MsgBox("Computer is not connected.")
End If
```


১২তম পরিচ্ছেদ: *Accessing Computer Resources*

কম্পিউটারের রিসোর্স সংগ্রহ ও ব্যবহার : *Accessing Computer Resources*

এবার বোতাম দুটির ক্লিক ইভেন্টে আমরা কোড লিখবো। প্রথমে ডবল ক্লিক করুন 'সব মুছে দিন' বোতামে। ক্লিক ইভেন্টে নিচের কোডটি লিখুন।

'clear text for new entry

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    TextBox1.Text = ""
    TextBox2.Text = ""
End Sub
```

উপরের কোডগুলো সহজেই অনুমেয়। আমরা উভয় টেক্সট বাক্স থেকে লেখা মুছে দিয়েছি। এবার ডবল ক্লিক করুন 'নতুন ফাইল ডাউনলোড করুন' বোতামে। ক্লিক ইভেন্টে লিখুন:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim FileURI As String = TextBox1.Text
    Dim FileSaveLoc As String = Label4.Text & TextBox2.Text
    If FileURI = "" Or TextBox2.Text = "" Then
        MsgBox("Please Enter URI address or File Name. One or Both are MISSING!!!",
        MsgBoxStyle.Exclamation)
        Exit Sub
    End If
    On Error GoTo ThisLine
    My.Computer.Network.DownloadFile(FileURI, FileSaveLoc, "", "", True, 500, True)
    MsgBox("File has been downloaded at: " & FileSaveLoc)
    Exit Sub
ThisLine:
    MsgBox("Sorry! An error occured. Please Try again. Make sure to enter correct texts
into fields.", MsgBoxStyle.Information)
End Sub
```

দ্বিতীয় এ কোড ব্লকটিই মূল কোড। এগুলো লাইন বাই লাইন বুঝিয়ে বলছি। প্রথমত আমরা টেক্সটবাক্স-১ এ যে টেক্সটটুকু আছে তা একটি স্ট্রিং ভেরিয়েবলে সংরক্ষণ করেছি। অনুরূপ আরেকটি ভেরিয়েবলে সংরক্ষণ করেছি লেবেল৪ ও টেক্সট২ এর লেখাটুকু। লক্ষ করুন আমরা নির্দেশ দিচ্ছি লেবেলের টেক্সটের সাথে টেক্সটবাক্স২ এর টেক্সট যোগ করতে। উভয়টি মিলে ফাইল সংরক্ষণের পূর্ণাঙ্গ প্যাথ সংরক্ষিত হবে FileSaveLoc ভেরিয়েবলে। ব্যবহারকারী কোনো একটি বা উভয় টেক্সট বাক্সে কিছু না লিখলে এয়ারর আসবে। তাই আমরা If - End If ব্লকে নিশ্চিত করছি ব্যবহারকারী উভয় ফিল্ডে টেক্সট লিখেছেন। না লিখলে একটি ম্যাসেজ আসবে। এরপর কোড রান করা এখানেই শেষ হবে।

On Error GoTo ThisLine -একটি পুরাতন কোডিং প্রাকটিস যা এখনো জনপ্রিয়। এখানে যা বলা হচ্ছে তাহলো: পরের কোড লাইনটিতে কোনো এয়ারর হলে ThisLine: -নামক কোডে জাম্প দাও। সেখানে যেয়ে একটি মেসেজ বাক্সে ভুলের রিপোর্ট প্রদর্শিত হবে। এরপর কোড রানিং শেষ হবে। এভাবে করার কারণ হলো, যাতে করে এয়াররের ফলে পুরো এ্যাপলিকেশন বন্ধ বা ক্র্যাশ না করে।

যাক, যে কোড লাইনটি আসল কাজ তথা ফাইল ডাউনলোড করে তাহলো:

```
My.Computer.Network.DownloadFile(FileURI, FileSaveLoc, "", "", True, 500, True)
```

আমরা ইতোমধ্যে জেনেছি এই মেথডের ৭টি আর্গুমেন্ট মানে কি। সুতরাং এখানে আর পুনরাবৃত্তি করছি না। তবে এটুকু বলছি যে, ব্যবহারকারী কর্তৃক লিখিত বা ওয়েবসাইট থেকে পেস্ট করা ফাইল হলো FileURI। অনুরূপ ব্যবহারকারী কর্তৃক নির্দেশিত ফাইল নামসহ সংরক্ষণের পূর্ণ প্যাথ হলো FileSaveLoc।

১৩তম পরিচ্ছেদ: Software Development Cycle



সফটওয়্যার তৈরির প্রক্রিয়াচক্র : Software Development Cycle



ব্যবহারযোগ্য বা কমার্শিয়াল সফটওয়্যার তৈরি করতে প্রোগ্রামাররা একটি সুচিন্তিত প্রণালীবদ্ধ পদ্ধতি অনুসরণ করে থাকেন। এতে আছে বেশ ক'টি পদক্ষেপ। সঠিকভাবে এগুলো অনুসরণ করলে কাজিত ফলাফল পাওয়ার নিশ্চয়তা থাকে। আমরা এরূপ ৬টি পদক্ষেপ সনাক্ত করেছি যাকে, বলা যায়: সফটওয়্যার তৈরির প্রক্রিয়াচক্র (বায়ের চিত্রটি দেখুন)। ইংরেজিতে বলতে পারি: *Software Development Cycle*।

পরের দীর্ঘ পরিচ্ছেদে আমরা বেশ ক'টি ব্যবহারযোগ্য সফটওয়্যার ডেভেলপ করবো। শিক্ষার্থীরা বায়ের চিত্রে প্রদর্শিত সব ক'টি পদক্ষেপ নিয়মতান্ত্রিকভাবে অনুসরণ করবেন- এটাই কাম্য। প্রত্যেক সফটওয়্যার তৈরির প্রক্রিয়া খুব স্পষ্ট ভাষায় ধারাবাহিকভাবে বর্ণিত হবে অবশ্যই। প্রথম প্রজেক্ট শুরু করার আগে আসুন, আমরা এ ৬টি পদক্ষেপের ওপর আরো কিছু ব্যাখ্যা তুলে ধরি।

ক. প্রয়োজন সনাক্তকরণ - Identify Requirement

প্রত্যেক সফটওয়্যার সৃষ্টি করা হয় কোনো এক বা একাধিক প্রয়োজন মেটানোর উদ্দেশ্যে। যেমন: মাইক্রোসফটের অফিস ২০০৭ সফটওয়্যারের মধ্যে মোট ৯টি ভিন্ন প্রোগ্রাম আছে। এগুলো হলো: Access, Excel, Groove, InfoPath, OneNote, Outlook, PowerPoint, Publisher and Word। এদের প্রত্যেকটির কাজ ভিন্ন। অফিস ডেভেলপমেন্ট প্রোগ্রামাররা কম্পিটারের সাহায্যে বিভিন্ন কাজ কিভাবে আঞ্জাম দেওয়া যায় তা সনাক্ত করেছেন। কোন প্রয়োজন কোন প্রোগ্রাম মেটাতে তা সনাক্তকরণের নামই হলো Identify Requirement। আমাদের ১ম ব্যবহারযোগ্য প্রোগ্রামের নাম: 'বাংলাদেশ শহর থেকে শহর দূরত্ব' [BD City to City Distance]। এটি সৃষ্টির পূর্বে প্রয়োজন সনাক্তকরণ করতে যেয়ে আমরা দেশের ৬৪টি জিলা শহরের মধ্যে দূরত্ব কিলোমিটার ও মাইলের হিসাবে জানার স্পৃহার কথা ভেবেছি। সুতরাং এ প্রয়োজন মেটানোর জন্যই আমরা প্রোগ্রাম ডেভেলপমেন্টে অগ্রসর হয়েছি।

খ. বিশ্লেষণ - Analysis

প্রয়োজন সনাক্তকরণের পর বুঝতে হবে কিভাবে তা মেটানো যায়। এজন্য প্রয়োজন সঠিক ব্যাখ্যার। আর এ পদক্ষেপের নামই হলো বিশ্লেষণ। আমাদের ১ম প্রোগ্রামের বিশ্লেষণ হলো: ব্যবহারকারী যে কোনো দু'টি শহরের নাম চয়েজ করে জেনে নেবেন এদের মধ্যে রোডের দূরত্ব কী। দূরত্বটি হবে কিলোমিটার ও মাইলের এককে।

গ. আকল্প - Design

উপরের উভয় পদক্ষেপগুলো থেকে সঠিক জবাব পাওয়ার পরই আসল ডেভেলপমেন্ট প্রক্রিয়া শুরু হবে। প্রথমে অবশ্যই ইউজার ইন্টারফেইস তৈরি করতে হবে। আর এ ইউআই সৃষ্টির নামই হলো প্রোগ্রাম ডিজাইন পদক্ষেপ বা আকল্প। আমাদের ১ম প্রোগ্রামের ডিজাইনটি দেখলেই বুঝতে পারবেন এ স্টেজে আসলে কী করতে হয়। একটু পরই তা দেখতে পারবেন।

১৪তম পরিচ্ছেদ: *Complete Application Tutorials*



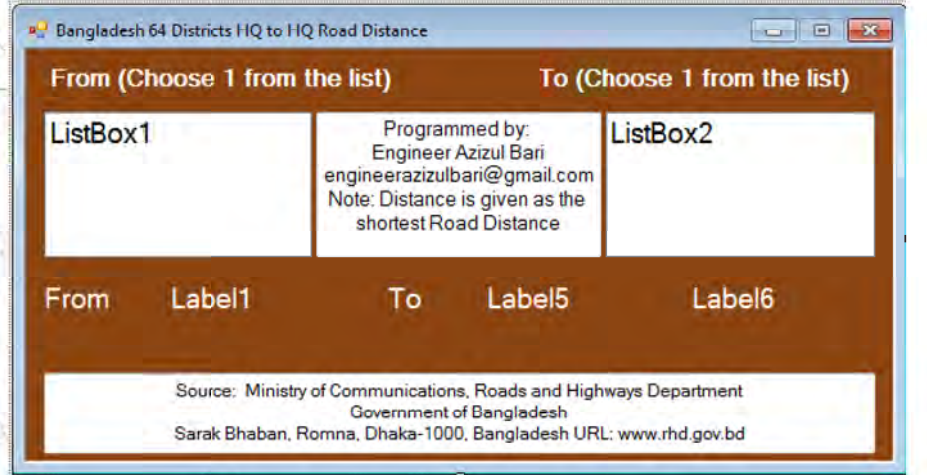
পূর্ণাঙ্গ এ্যাপলিকেশন প্রশিক্ষণ : *Complete Application Tutorials*

Application 1: BD City to City Distance - বাংলাদেশ শহর থেকে শহর দূরত্ব

উদ্দেশ্য: আমাদের উদ্দেশ্য হলো, ব্যবহারকারী বাংলাদেশের ৬৪টি জিলা শহরের রোড-দূরত্ব সহজে জানতে পারবেন। দূরত্বে প্রদর্শিত হবে কিলোমিটার ও মাইল এককে।

ডিজাইন: প্রথমেই আমরা একটি আকর্ষণীয় ইউজার ইন্টারফেইস ডিজাইন করবো। বায়ের চিত্রটি দেখুন ও পদক্ষেপগুলো অনুসরণ করুন।

১. একটি নতুন প্রজেক্ট তৈরি করুন। নামকরণ করুন: “BD City to City Distance”।
২. বায়ের চিত্রের মতো ফর্মটি ডিজাইন করুন।



ফর্মের বস্তুসমূহ: ৭টি লেবেল, ২টি লিস্ট-বাক্স এবং (চাইলে) দু’টি টেক্স-বাক্স।

ফর্মটি যাতে একই মাপের থাকে সেজন্য এর `FormBorderStyle = Fixed3D`, `MaximizeBox = False` and `AutoScaleMode = None` হবে। এর `Text` প্রপার্টি হবে `Bangladesh 64 Districts HQ to HQ Road Distance`। আপনি চাইলে অন্য নামও দিতে পারেন। রং আমি একটা চয়েজ করেছি। আপনি যা চান সেটাই দিতে পারেন। সবশেষে দু’টি ফিল্ডে কিছু তথ্য দিয়েছি। নিচেরটি জরুরী বলে মনে করি। আর উপরেরটি না দিলেই হবে। অবশ্য থাকলে তো আপনার নিজের নামই থাকবে, আমার পরিবর্তে! বিভিন্ন লেবেলে যা লিখা আছে তা-ই (স্টেট) প্রোপার্টিতে লিখুন। ব্যস! ডিজাইন হয়ে গেলো। এবার আসল কাজ- কোডিং!

প্রোগ্রামের তথ্যাদি: আমাদের এই বিশেষ প্রোগ্রামটি কিছু তথ্যের উপর নির্ভরশীল। আগে থেকেই আমাদেরকে ৬৪টি জিলার নামসহ একটা থেকে আরেকটির দূরত্ব (কিলোমিটার এককে) জানতে হবে। এজন্য আমরা যোগাযোগ মন্ত্রণালয়ের ওয়েবসাইটটি কাজে লাগিয়েছি। সেখানে দেওয়া আছে একটি টেবিল। এতে জিলাগুলোর নাম ও একটা থেকে আরেকটির দূরত্ব মেট্রিক্স পদ্ধতিতে লিখিত আছে। এই তথ্যগুলোর উপর নির্ভর করেই আমরা পুরো এ্যাপলিকেশনটি তৈরি করেছি। একজন ব্যবহারকারী প্রশ্ন করতে পারেন: দূরত্ব আপনি পেলেন কোথায়? এগুলো কি সঠিক? আমাদেরকে জবাব দিতে হবে না। বাংলাদেশের তথ্য মন্ত্রণালয় থেকেই যে তথ্যগুলো পেয়েছি তা-তো আমরা নিচের টেক্স বাক্সে ঘোষণা দিয়েছি! যাক এবার আসুন কোডগুলোর ওপর আলোচনা শুরু করি। আপনি শহরের নাম ও দূরত্ব সংখ্যা টাইপ করতে সতর্কতা অবলম্বন করবেন- তথ্যে ভুল হলে হয় প্রোগ্রাম কাজ করবে না, কিংবা ভুল ফলাফল পাবেন। আমরা এর কোনোটাই চাই না।

২. কোডিং: ফর্মে ডবল-ক্লিক করে কোড জানালা খুলুন। প্রথমেই আমাদেরকে কয়েকটি গ্লোব্যাল ভেরিয়েবল ডিক্লেয়ার করতে হবে। নিচের (পরের পৃষ্ঠায় দেওয়া) কোড লাইন মুতাবিক এগুলো ডিক্লেয়ার করুন।

১৪তম পরিচ্ছেদ: *Complete Application Tutorials*

পূর্ণাঙ্গ এ্যাপলিকেশন প্রশিক্ষণ : *Complete Application Tutorials*

```

'=== special case Dhaka. ref point
If IndexA = 0 Then
    If IndexB = 0 Then 'Dhaka to Dhaka chosen
        Label6.Text = "is: " & 0 & " km"
        Label8.Text = "Equal: " & 0 & " Miles"
        Exit Sub
    End If
    Label6.Text = "is: " & FindDisanceZero(IndexB) & " km" 'If both condition results FALSE, this func. called
End If
'===== check from which city chosen =====
Label6.Text = "is: " & ManyatOnce() & " km" 'call function and display result
'===== get distance in miles and display'
Dim KMvalue As Integer = Mid(Label6.Text, 5, (Len(Label6.Text) - 7)) 'get km value to a variable as number
Label8.Text = "Equal: " & Math.Truncate(KMvalue * 0.621371) & " M" 'convert to miles and display
ListBox1.SelectedItem = ""
ListBox2.SelectedItem = ""
End Sub

```

লাইন বাই লাইন ব্যাখ্যা: প্রায় প্রতিটি লাইনের পাশে (সবুজ রঙে) কमेंট করেছি বুঝার জন্য। এরপরও মনে করছি বুঝিয়ে দেওয়া আপনার জন্য সুবিধে হবে।

ক. প্রথম চার লাইনে চারটি ভেরিয়েবল ডিক্লার করার করেছি। এর দু'টি ইন্টজার ও দু'টি স্ট্রিং টাইপ। এগুলোতে যথাক্রমে লিস্ট বাক্সের ইনডেক্স সংখ্যা ও সিলেক্টেড টেক্সট সংরক্ষিত থাকবে।

খ. পঞ্চম লাইন দ্বারা নিশ্চিত করা হয়েছে, ব্যবহারকারী উভয় লিস্ট বাক্সে কোনো শহরের নাম সিলেক্ট করেছেন। না করলে প্রোগ্রাম বন্ধ হয়ে যাবে। এমতাবস্থায় সাবরুটিন এক্সেকিউশন বন্ধ করা জরুরী- যা এই লাইনটি নিশ্চিত করে।

ঘ. এর পরের চারটি লাইন দ্বারা প্রথম চার লাইনে ডিক্লার করার ভেরিয়েবলগুলোতে ভ্যালু সংরক্ষণ করেছি। অর্থাৎ উভয় লিস্ট বাক্সের ইনডেক্স সংখ্যা ও শহরের নাম ভেরিয়েবলে নিয়ে এসেছি। আগের পৃষ্ঠার শেষ লাইনটি দ্বারা আমরা দ্বিতীয় লিস্ট বাক্সের সিলেক্টেড টেক্সট তথা শহরের নাম লেবেলে এনে প্রদর্শন করেছি। এবার আমাদের কাজ হলো সঠিক দূরত্ব কিলোমিটার ও মাইল এককে প্রদর্শন করা।

ঙ. দূরত্ব প্রদর্শনের জন্য আমরা একই ইভেন্টই ব্যবহার করবো। অর্থাৎ ব্যবহারকারী দেখবেন দ্বিতীয় লিস্ট বাক্সে ক্লিক করলেই দূরত্ব প্রদর্শিত হচ্ছে। এ কাজের জন্য প্রথমত বিশেষ কেইস বিবেচনা করতে হবে- তাহলো ঢাকা থেকে ঢাকা। উভয়টি সিলেক্ট হলে রেজাল্ট হবে ০। আমরা এ কাজের জন্য দু'টি ইফ-এন্ড ইফ [If - End If] ব্লক ব্যবহার করে চেক করেছি। দ্বিতীয় ইফ-এন্ড ইফ প্রথমটির ভেতরে নেস্ট করা হয়েছে। অন্যকথায়, যদি উভয় ইনডেক্স = ০ হয় তাহলে বুঝতে হবে ঢাকা থেকে ঢাকা চয়েজ করা হয়েছে। আর এর ফলাফল ০। সুতরাং ডিসপ্লেতে ০ প্রদর্শন করে প্রোগ্রাম এখানেই থেমে যেতে হবে। এজন্যই লেবেল ৬ ও ৮-এ ০ প্রদর্শন করে Exit Sub কোড দ্বারা সাবরুটিন থেকে বেরিয়ে আসার নির্দেশ দেওয়া হয়েছে।

চ. পরের লাইনেই মূলত দূরত্ব প্রদর্শনের নির্দেশ দেওয়া হয়েছে। এতে বলা আছে: লেবেল৬ এর টেক্সট হবে:

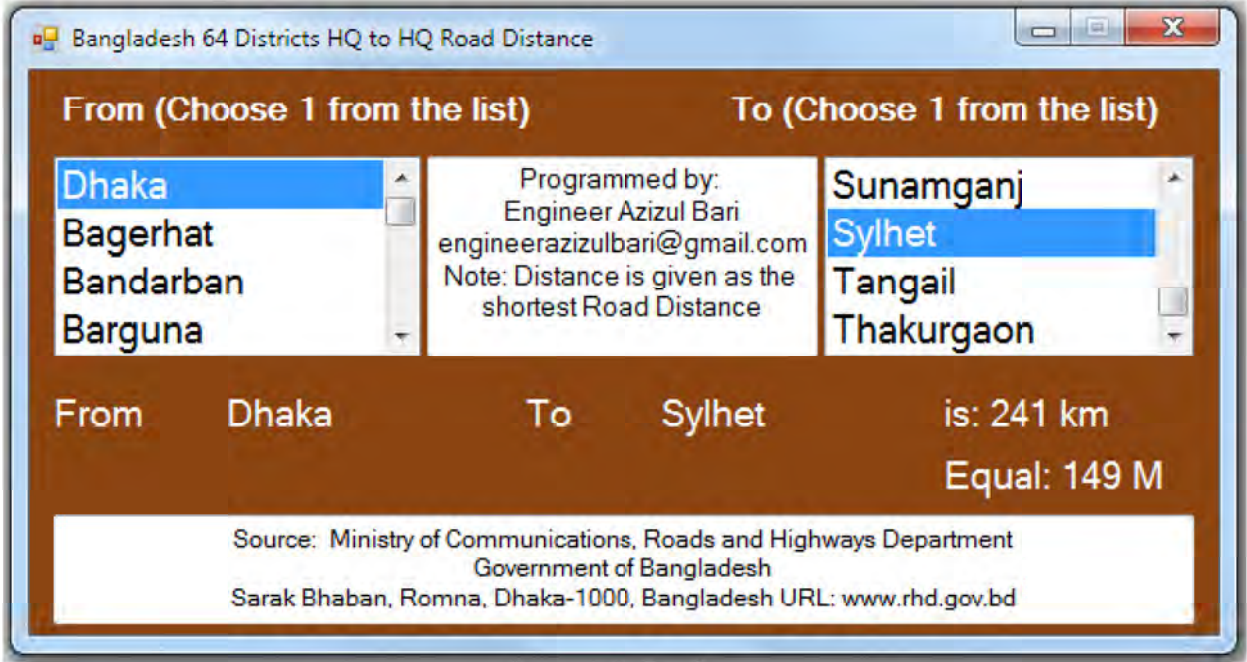
"is: " & ManyatOnce() & " km" - যার অর্থ, উভয়দিকে দু'টি শব্দ [is: , km] থাকবে এবং মাঝে লিখা হবে দূরত্ব যা আসবে [ManyatOnce()] ফাংশনের ফলাফল থেকে।

ছ. পরের দু'টি লাইনের কাজ হলো কিলোমিটার একককে পরিবর্তন করে মাইল এককে রূপান্তর ও লিস্ট বাক্স ৮-এ তা প্রদর্শন করা। সুতরাং ব্যবহারকারী কিলোমিটার ও মাইল এ উভয় এককে দূরত্ব জানতে পারবেন। সর্বশেষ দু'টি লাইন দ্বারা উভয় লিস্ট বাক্সের আইটেম সিলেকশন বাতিল করা হয়েছে। ফলে পরবর্তীতে ভুল ফলাফল আসার সম্ভাবনা আর থাকবে না।

জ. এবার শেষের ফাংশন সম্পর্কে কিছু বলার আছে। প্রথমে এটি কী তা [পরের পৃষ্ঠায়] দেখে নিই।

১৪তম পরিচ্ছেদ: *Complete Application Tutorials*

পূর্ণাঙ্গ এ্যাপলিকেশন প্রশিক্ষণ : *Complete Application Tutorials*



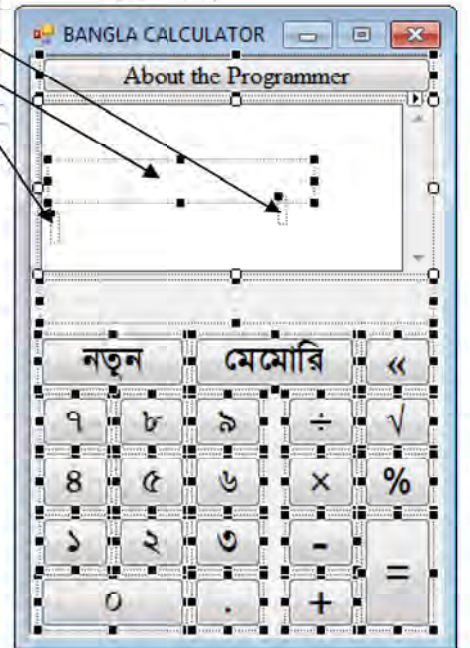
Application 1 -চলাকালীন ঢাকা থেকে সিলেটের দূরত্ব কিলোমিটার ও মাইল এককে প্রদর্শিত হচ্ছে।

Application 2 : Bangla Calculator - বাংলা ক্যালকুলেটর

উদ্দেশ্য: প্রশ্ন থাকতে পারে, বাংলা ক্যালকুলেটর- কেনো? জবাব- কেনো না? তবে আসল জবাব হলো- আমরা তো শিখতে চাই! ক্যালকুলেটর কোডিং কিভাবে করতে হয় তা জানতে চাই। আর ডিসপ্লে বাংলা সংখ্যায় হলে তো ভালোই হলো! সুতরাং আসুন, গড়ে তুলি পিসি'র জন্য একটি উন্নত বাংলা ক্যালকুলেটর এ্যাপলিকেশন।

ডিজাইন: ডানের ডিজাইনটি দেখুন। আমাদের ক্যালকুলেটরে সাধারণ পাটিগণিতের চারটি ফাংশন [+ , - , ÷ , x] ছাড়াও থাকবে পারসেন্ট (%) এবং বর্গমূল ($\sqrt{\quad}$)।

লুকানো
লেবেল



উপরে আছে প্রোগ্রামার সম্পর্কে তথ্য প্রদর্শনের একটি বোতাম [About the Programmer]। এর পরে আছে মূল প্রদর্শন জানালা [TextBox1]। এ বাক্সেই সংখ্যা প্রবেশকরণ ও প্রদর্শন ক্রিয়া সম্পাদিত হবে। এটাই হলো, মূল ডিসপ্লে। আমরা তিনটি লুকানো [label2, label3, label4] লেবেল দেখছি। এগুলোতে কিছু তথ্য সংরক্ষিত থাকবে যা ব্যবহারকারী দেখার প্রয়োজন নেই। সুতরাং এদের ভিজিবল প্রোপার্টি হবে: False। ডিসপ্লে'র একটু নিচে লম্বাটে একটি লেবেল [label1] আছে। এটা দ্বারা ব্যবহারকারীকে কিছু তথ্য প্রদর্শন করা হবে। বাকী সবই বোতাম। উপরের সারির তিনটি বোতামের কাজ যথাক্রমে: নতুন হিসাব, সংরক্ষিত সংখ্যা দেখা ও ডিসপ্লেতে সর্বশেষ সংখ্যা মেটানো। বাকি সব বোতামের কাজ কি শিক্ষার্থীরা সহজেই অনুমান করতে পারছেন। সুতরাং আর বেশি বলছি না। তবে মনে রাখবেন, বোতামের টেক্সট [Text] প্রোপার্টিতে যা দেখা যাচ্ছে তা চুকাতে হবে। অর্থাৎ ১, ২, ০, ., =, %, $\sqrt{\quad}$ ইত্যাদি হবে বোতামগুলোর টেক্সট প্রোপার্টি।

১৪তম পরিচ্ছেদ: *Complete Application Tutorials*

পূর্ণাঙ্গ এ্যাপলিকেশন প্রশিক্ষণ : *Complete Application Tutorials*

'Times Button

```
Private Sub ButTimes_Click(sender As Object, e As EventArgs) Handles ButTimes.Click
    If Label3.Text <> "" Then Exit Sub
    If DisplayNotEmpty() = False Then
        Label1.Text = TextBox1.Text & ",Y" : Label4.Text = "x"
        FirstValue()
    End If
End Sub
```

'Divide Button

```
Private Sub ButDivide_Click(sender As Object, e As EventArgs) Handles ButDivide.Click
    If Label3.Text <> "" Then Exit Sub
    If DisplayNotEmpty() = False Then
        Label1.Text = TextBox1.Text & "-/" : Label4.Text = "÷"
        FirstValue()
    End If
End Sub
```

'percentage Button

```
Private Sub ButPercent_Click(sender As Object, e As EventArgs) Handles ButPercent.Click
    If Label3.Text <> "" Then Exit Sub
    If DisplayNotEmpty() = False Then
        Label1.Text = TextBox1.Text & "% " : Label4.Text = "%"
        FirstValue()
    End If
End Sub
```

উপরের প্রতিটি সাবের কাজ যেহেতু একই ধরনের তাই একটির ওপর ব্যাখ্যা তুলে ধরাই যথেষ্ট হবে। সাবের প্রথম লাইন দ্বারা নিশ্চিত করা হচ্ছে ইতোমধ্যে ব্যবহারকারী কোনো সংকেতে ক্লিক করে নেন নি। যদি করে থাকেন তাহলে তা লেবেলে [Label3 -তে] সংকেত চুকানো হয়েই গেলো। সুতরাং [Exit Sub -দ্বারা] সাব থেকে বের হতে হবে।

দ্বিতীয় চেকটি হলো ডিসপ্লেতে সংখ্যা আছে কি না তা জানা। সংখ্যা থাকলেই [ফাংশনের ফলাফল] পরের লাইনে যেয়ে আমরা প্রথমে লেবেল১-এ ব্যবহারকারীকে কিছু তথ্য দিচ্ছি এবং দ্বিতীয়ত গোপন লেবেল৪-এ সংকেতটি সংরক্ষণ করছি। এরপর ফাস্টভ্যালু নামক সাবরুটিন কল করেছি। যার কাজ হলো লেবেল৩-তে ডিসপ্লেট সংখ্যা নিয়ে সংরক্ষণ করা ও ডিসপ্লে মুছে দেওয়া। এই কাজগুলো উপরের ৫টি সাব দ্বারা সম্পাদিত হয়। এবার আমরা 'স্কোয়ার রুট' [বর্গমূল] ক্যালকুলেশনের বিশেষ সাবটির ওপর তথ্যাদি তুলে ধরছি।

'Square Root Calculation and display sub

```
Private Sub ButSqr_Click(sender As Object, e As EventArgs) Handles ButSqr.Click
    If TextBox1.Text <> "" And Label1.Text = "" Then 'make sure number exists
        Dim SquareRoot As Double = Math.Sqrt(TextBox1.Text) 'get sqr into variable
        Label1.Text = TextBox1.Text & " Gi eM@g~j" 'bangla info display
        TextBox1.Text = Math.Round(SquareRoot, 8) 'put result into display
        Label2.Text = SquareRoot 'put result into memory
    End If
End Sub
```

বুঝতেই পারছেন, বর্গমূল বের করা অন্যান্য সকল কাজ থেকে ভিন্ন। এর কারণ হলো, বর্গমূলের জন্য একটি মাত্র সংখ্যার প্রয়োজন। অপরদিকে +, -, ÷, X ও % বের করতে দু'টি সংখ্যা থাকা জরুরী। সুতরাং ওগুলোর ক্যালকুলেশনও হবে একই ধরনের।

১৪তম পরিচ্ছেদ: *Complete Application Tutorials*

পূর্ণাঙ্গ এ্যাপলিকেশন প্রশিক্ষণ : *Complete Application Tutorials*

```

If Label4.Text = "-" Then 'check which calculation needed
    CalValue = Val(Label3.Text) - Val(TextBox1.Text) 'actual calculation into variable
    Label1.Text = Label1.Text & " - " & TextBox1.Text 'Info display
    TextBox1.Text = Math.Round(CalValue, 8) 'display to 8 decimal places
    Label3.Text = "" 'clear hidden value
    Label2.Text = CalValue 'result into memory
    Exit Sub
End If
If Label4.Text = "x" Then 'check which calculation needed
    CalValue = Val(Label3.Text) * Val(TextBox1.Text) 'actual calculation into variable
    Label1.Text = Label1.Text & " x " & TextBox1.Text 'Info display
    TextBox1.Text = Math.Round(CalValue, 8) 'display to 8 decimal places
    Label3.Text = "" 'clear hidden value
    Label2.Text = CalValue 'result into memory
    Exit Sub
End If
If Label4.Text = "/" Then 'check which calculation needed
    'check if user trying to divide by 0
    If Val(TextBox1.Text) = 0 Then
        MsgBox("আহ! ০-দ্বারা ভাগ করা তো বৈধ নয়!") 'if so show a message and exit
    End If
    CalValue = Val(Label3.Text) / Val(TextBox1.Text) 'actual calculation into variable
    Label1.Text = Label1.Text & " / " & TextBox1.Text 'Info display
    TextBox1.Text = Math.Round(CalValue, 8) 'display
    Label3.Text = "" 'clear hidden value
    Label2.Text = CalValue 'result into memory
    Exit Sub
End If
If Label4.Text = "%" Then 'check which calculation needed
    CalValue = (Val(Label3.Text) / 100) * Val(TextBox1.Text) 'actual calculation into variable
    Label1.Text = TextBox1.Text & " % " & Label1.Text 'Info display
    TextBox1.Text = Math.Round(CalValue, 8) 'display to 8 decimal places
    Label3.Text = "" 'clear hidden value
    Label2.Text = CalValue 'result into memory
    Exit Sub
End If
End Sub

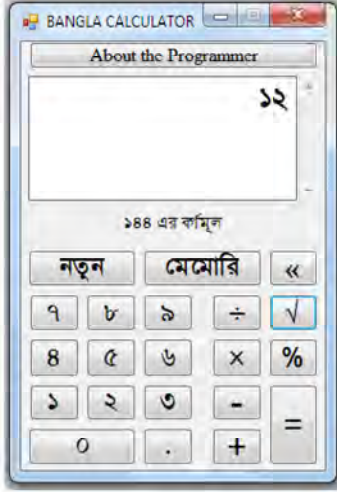
```

প্রথমেই মনে করিয়ে দিচ্ছি, এ সাবরুটিন হলো সমান চিহ্নের বোতামের ক্লিক ইভেন্ট। প্রথমেই সবকিছু ঠিকমতো ডিসপ্লেতে চুকানো হয়েছে কি না তা দেখতে হবে। এ কাজের জন্য একটি ফাংশন বানিয়েছি- যার বর্ণনা শেষে আসবে। এ ফাংশনের কাজ হলো 'সত্য' কিংবা 'মিথ্যা' ফলাফল রিটার্ন করা। আমাদের উদ্দেশ্য সত্য ফলাফল পাওয়া। না পেলে সাবরুটিন থেকে বের হয়ে যেতে হবে। ফলাফল সত্য হলে পরের লাইনে যেয়ে একটি ডাবল টাইপ ভেরিয়েবল [CalValue] ঘোষণা করেছি যাতে থাকবে ফলাফল সংখ্যাটি। পরের লাইনে যেয়ে চেক করা হয়েছে কোন্ ক্যালকুলেশন হবে [+ যোগ হবে কি?], কথা সত্য হলে, এ সময় অবশ্যই লুকানো লেবেল৪-এ '+' লিখা থাকবে। ধরে নিন + লিখা আছে। এবার যোগ অঙ্ক কষতে হবে। পরের লাইনে আগে ঘোষিত ভেরিয়েবলে সংখ্যা দু'টি যোগ করে সংরক্ষণ করা হয়েছে। এরপর লেবেল১-এ কিছু দৃশ্যমান তথ্য দেওয়া হয়েছে, ব্যবহারকারী কোন্ সংখ্যার সঙ্গে কোন্ সংখ্যা যোগ দিলেন তা বুঝানোর জন্য। পরের লাইন দ্বারা যোগফলটি ৮টি ডেসিমেল প্লেইস পর্যন্ত প্রদর্শনের জন পরিবর্তন করে ডিসপ্লে ফিন্ডে দেখানো হয়েছে। পরের দু'টি লাইনের প্রথমটি দ্বারা লুকানো লেবেল-৩ এর মান মুছে ফেলার পর দ্বিতীয় লাইন দ্বারা লুকানো লেবেল২-এ প্রাপ্ত মানটি মেমোরি হিসাবে সংরক্ষিত হবে।

১৪তম পরিচ্ছেদ: Complete Application Tutorials

পূর্ণাঙ্গ এ্যাপলিকেশন প্রশিক্ষণ : Complete Application Tutorials

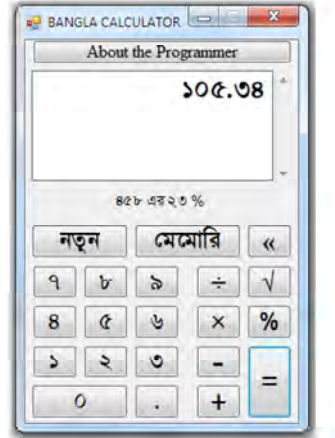
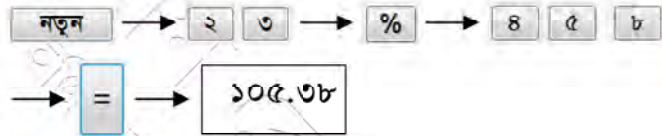
পরীক্ষণ ও বিস্তারণ: বাংলা ক্যালকুলেটর এ্যাপলিকেশনের প্রোগ্রামিং বা কোডিং পর্যায় শেষ হলো। এবার একে পরীক্ষা করতে হবে। এজন্য প্রয়োজন বিন্দু করে পরিচালনা। তাই এক্ষুণি পরীক্ষা শুরু হয়ে যাক। নিচের চিত্রে চলাকালীন জানালাটি দেখতে পাচ্ছেন।



আমরা মূল ডিসপ্লে ফিল্ডে সংখ্যা ১২ দেখতে পাচ্ছি। ডিসপ্লে নিচে আছে '১৪৪ এর বর্গমূল' কথাটি। সুতরাং বুঝাই যাচ্ছে, ১৪৪ এর বর্গমূল হলো ১২- এই ক্যালকুলেটর করেছি। এটা করতে প্রথমে ক্যালকুলেটরে আপনি যেভাবে সংখ্যা ঢুকাবেন তাহলো:



অনুরূপ ৪৫৮ -এর ২৩ শতাংশ বের করতে আপনি যে ভাবে সংখ্যা ঢুকাবেন তাহলো (ডানের চিত্র):



এ দু'টো ফাংশনের কাজ দেখানোর কারণ হলো, এগুলোই আমাদের বাংলা ক্যালকুলেটরের বিশেষ ফাংশন। বাকি চারটে পাটিগণিতের ফাংশন [+ , - ÷ ও x] সচরাচর ব্যবহৃত ক্যালকুলেটরের মতোই কাজ করে। সুতরাং ওগুলোর অপারেশন এখানে দেখাচ্ছি না। তবে আপনার উচিত হবে সবগুলো ফাংশনই একে এক পরীক্ষা করা। আপনার এ সৃষ্টিকে তো শেষ পর্যন্ত গ্রাহকদের নিকট পৌঁছানোর দরকার! আর এ কাজের নামই বিস্তারণ।

ভিজুয়াল স্টুডিও এক্সপ্রেস ফর ডেস্কটপ পূর্ণাঙ্গ এ্যাপলিকেশন বিস্তারণে আমাদেরকে বিশেষ সহযোগিতা করে। টেস্টিং স্তরের কাজ সম্পন্ন হওয়ার পর আপনি সহজেই কয়েকটির পদক্ষেপে সফটওয়্যার ডিস্ট্রিবিউটের জন্য ইনস্টলার তৈরি করতে পারেন। আমরা গ্রন্থের শেষে ইনস্টলার তৈরির পদ্ধতির ওপর বিস্তারিত আলোচনা করবো। এবার আসুন একটি উন্নতমানের বাংলা / ইংরেজি একক পরিবর্তন সফটওয়্যার তৈরি করি! এ প্রোগ্রামটি দীর্ঘ, সময়সাপেক্ষ এবং কিছুটা জটিল। তবে প্রোগ্রাম করা শেষ হলে আপনি অবশ্যই আনন্দিত হবেন। কবি বলেন নি, এ সৃষ্টি সুখের উল্লাসে!

Application 3 : Bangla / English Unit Conversion Software - বাংলা / ইংরেজি একক পরিবর্তন সফটওয়্যার।

সফটওয়্যার তৈরির লক্ষ্য: আমরা একটি বাংলা / ইংরেজি একক পরিবর্তন সফটওয়্যার তৈরি করবো যাতকরে ব্যবহারকারী অতি সহজে বিভিন্ন এককের মধ্যে সম্পর্ক, পরিবর্তন, সাংখ্যিক ফলাফল ইত্যাদি বের করতে সক্ষম হন।

ব্যাপ্তি: বিভিন্ন এককের মধ্যে শত শত সম্পর্ক বিদ্যমান। তবে আমাদের টার্গেট হলো সচরাচর ব্যবহৃত এককগুলো নিয়ে কাজ করা। সেমতে আমরা নিম্নের [পরের পৃষ্ঠার] টেবিলে দেওয়া ইউনিটগুলো নিয়ে কাজ করবো।

১৪তম পরিচ্ছেদ: *Complete Application Tutorials*

পূর্ণাঙ্গ এ্যাপলিকেশন প্রশিক্ষণ : *Complete Application Tutorials*

৭. আমাদের সৃষ্ট সাব: MileMileKilo() -এটা ফর্মের লোড [Form1_Load] সাব থেকে কল করা হয়েছে।
৮. আমাদের সৃষ্ট সাব: Combo1One() -এটা সিস্টেম ইভেন্ট সাব ComboBox1_SelectedIndexChanged - থেকে কল করা হয়েছে।
৯. আমাদের সৃষ্ট সাব: AreaMileKilo() -এটা সিস্টেম ইভেন্ট সাব ComboBox1_SelectedIndexChanged - থেকে কল করা হয়েছে।
১০. আমাদের সৃষ্ট সাব: TempDegFahren() -এটা সিস্টেম ইভেন্ট সাব ComboBox1_SelectedIndexChanged - থেকে কল করা হয়েছে।
১১. আমাদের সৃষ্ট সাব: WeightKiloSher() -এটা সিস্টেম ইভেন্ট সাব ComboBox1_SelectedIndexChanged - থেকে কল করা হয়েছে।
১২. আমাদের সৃষ্ট সাব: SurveyFunction() -এটা সিস্টেম ইভেন্ট সাব ComboBox1_SelectedIndexChanged - থেকে কল করা হয়েছে।
১৩. আমাদের সৃষ্ট সাব: CalculateAndDisplayLength() -এটার কাজ হলো 'দূরত্ব' ক্যাটাগরির কোন একক থেকে কোন একক পরিবর্তন হবে তা সনাক্ত করে মডিউলের ফাংশন কল করা। এরপর ফাংশন থেকে প্রাপ্ত ফলাফল ডিসপ্লে ফিল্ড টেক্সট বাক্স ২-এ প্রদর্শন করা। এ সাবটি উপরের (১৩ নং) সিস্টেম সাব থেকে কল করা হয়।
১৪. আমাদের সৃষ্ট সাব: CalculateAndDisplayArea() -এটাও উপরোক্ত সাবের মতো কাজ করে। পার্থক্য হলো, এর কাজ 'ক্ষেত্র' ক্যাটাগরির সঙ্গে সম্পৃক্ত।
১৫. আমাদের সৃষ্ট সাব: CalculateAndDisplayTemp() -এটাও উপরোক্ত সাবের মতো কাজ করে। পার্থক্য হলো, এর কাজ 'তাপ' ক্যাটাগরির সঙ্গে সম্পৃক্ত।
১৬. আমাদের সৃষ্ট সাব: CalculateAndDisplayWeight() -এটাও উপরোক্ত সাবের মতো কাজ করে। পার্থক্য হলো, এর কাজ 'ওজন' ক্যাটাগরির সঙ্গে সম্পৃক্ত।
১৭. আমাদের সৃষ্ট সাব: CalculateAndDisplaySurvey() -এটাও উপরোক্ত সাবের মতো কাজ করে। পার্থক্য হলো, এর কাজ 'সার্ভে' ক্যাটাগরির সঙ্গে সম্পৃক্ত।

উপরোক্ত ১৭টি সাবরুটিনের সবই প্রোগ্রামের মেইন ক্লাস সাবরুটিন তথা **Public Class Form1** -এর ভেতর অবস্থান করে। আগেই বলেছি দু'টি মডিউল আমরা নিজেই ডিফাইন করেছি আলাদাভাবে। এগুলোর ভেতর বেশ কিছু সাব ও ফাংশন আছে। মডিউল দু'টোর ব্যাপ্তি 'পাবলিক' হওয়ায় আমরা এসব সাব ও ফাংশন ফর্মের ক্লাস থেকে কল করার সুযোগ পেয়েছি। এ পর্যায়ে মডিউলে ডিফাইন করা প্রতিটি সাব সম্পর্কে বিস্তারিত বলার প্রয়োজন মনে করছি না। পরের পৃষ্ঠাগুলোতে আমরা সকল কোড একে একে তুলে ধরেছি। প্রথমে এসেছে **Public Class Form1** -এর ভেতরস্থ সকল কোড। বিস্তারিত ব্যাখ্যা শেষে যথাক্রমে **Public Module AllDisplayFunctionsModule** -এর ভেতরস্থ কোড ও **Public Module CalculationModule** -এর ভেতরস্থ সকল কোড তুলে ধরেছি। প্রত্যেকটির ওপর বিস্তারিত ব্যাখ্যা তো অবশ্যই আছে। শিক্ষার্থীদের ধৈর্যধারণ ক্ষমতা আরো বেশি থাকা চাই। পুরো কোড তুলে ধরতে বইয়ের ১৫ পৃষ্ঠা লেগেছে। কোডের প্রতিটি লাইন ব্যাখ্যার আলোকে পাঠ করে বুঝতে হবে। সুতরাং আসুন, এবার পুরো কোড দেখে নিই ও বুঝার চেষ্টা করি। এই কোডটি কিভাবে কাজ করে তা যদি আমরা সঠিকভাবে বুঝতে সক্ষম হই তাহলে ভিজুয়াল বেসিক.নেট প্রোগ্রামিং সম্পর্কে তো বেশ দূরে অগ্রসর হয়ে গেলাম! তাই না? অবশ্য পরের প্রোগ্রামটি আরো একটু কঠিন! সুতরাং ধৈর্যের মাত্রা তখন আরো বাড়তে হবে।

১৪তম পরিচ্ছেদ: *Complete Application Tutorials*

পূর্ণাঙ্গ এ্যাপলিকেশন প্রশিক্ষণ : *Complete Application Tutorials*

```
Public Function CalculatedValueSqMileTo(EnteredNumber As Double, ComBox3SelItemIndex As Integer) As Double
    Dim ForCounter As Integer
    For ForCounter = 0 To 6
        If ForCounter = ComBox3SelItemIndex Then
            Return AreaRelation8(ForCounter) * EnteredNumber
        End If
    Next
    Return 0
End Function
```

```
Public Function CalculatedValueSqKilometreTo(EnteredNumber As Double, ComBox3SelItemIndex As Integer) As Double
    Dim ForCounter As Integer
    For ForCounter = 0 To 6
        If ForCounter = ComBox3SelItemIndex Then
            Return AreaRelation9(ForCounter) * EnteredNumber
        End If
    Next
    Return 0
End Function
```

```
Public Function CalculatedValueSqMetreTo(EnteredNumber As Double, ComBox3SelItemIndex As Integer) As Double
    Dim ForCounter As Integer
    For ForCounter = 0 To 6
        If ForCounter = ComBox3SelItemIndex Then
            Return AreaRelation10(ForCounter) * EnteredNumber
        End If
    Next
    Return 0
End Function
```

This function returns Area, SqMillimetre to all combination calculated value. 3 parameters comes from calling sub

```
Public Function CalculatedValueSqMillimetreTo(EnteredNumber As Double, ComBox3SelItemIndex As Integer) As Double
    Dim ForCounter As Integer
    For ForCounter = 0 To 6
        If ForCounter = ComBox3SelItemIndex Then
            Return AreaRelation11(ForCounter) * EnteredNumber
        End If
    Next
    Return 0
End Function
```

```
Public Function CalculatedValueSqYardTo(EnteredNumber As Double, ComBox3SelItemIndex As Integer) As Double
    Dim ForCounter As Integer
    For ForCounter = 0 To 6
        If ForCounter = ComBox3SelItemIndex Then
            Return AreaRelation12(ForCounter) * EnteredNumber
        End If
    Next
    Return 0
End Function
```

This function returns Area, SqFt to all combination calculated value. 3 parameters comes from calling sub

```
Public Function CalculatedValueSqFtTo(EnteredNumber As Double, ComBox3SelItemIndex As Integer) As Double
    Dim ForCounter As Integer
    For ForCounter = 0 To 6
        If ForCounter = ComBox3SelItemIndex Then
            Return AreaRelation13(ForCounter) * EnteredNumber
        End If
    Next
    Return 0
End Function
```

This function returns Area, SqInch to all combination calculated value. 3 parameters comes from calling sub

```
Public Function CalculatedValueSqinchTo(EnteredNumber As Double, ComBox3SelItemIndex As Integer) As Double
    Dim ForCounter As Integer
```

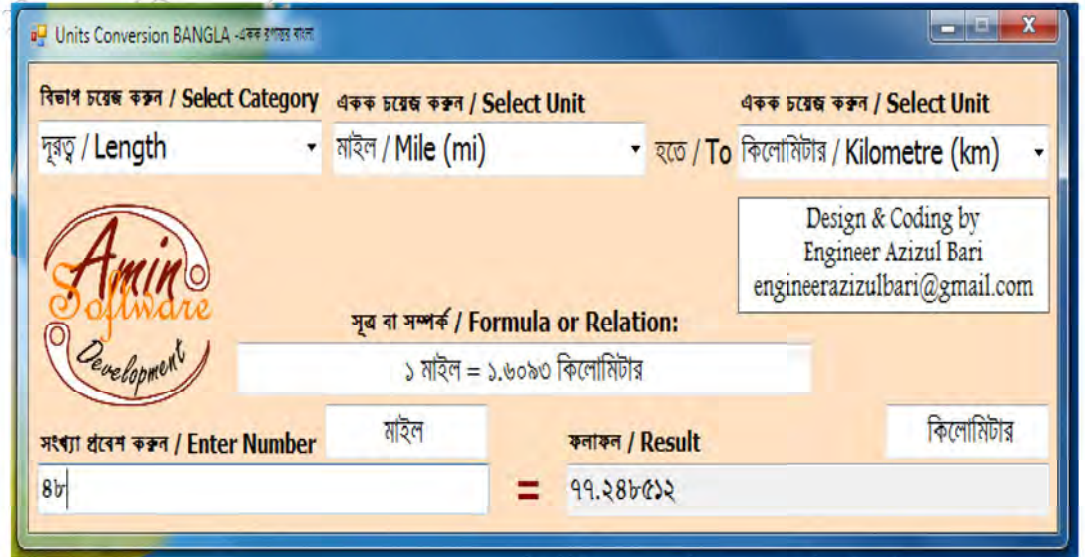



ভিজুয়াল বেসিক.নেট প্রোগ্রামিং

নতুন শিক্ষার্থীদের জন্য গুরুত্বপূর্ণ প্রশিক্ষণ রিসোর্স

দ্বিতীয় খণ্ড

Part Two



বিশেষ ইন্টারনেট সংস্করণ
Special Internet Edition

Copyright NOTICE:

All rights reserved. This electronic book or any portion thereof may not be reproduced or used in any manner whatsoever without the express written permission of the author except for the use of brief quotations in a book review. This electronic book may only be used for private reading, it is forbidden to print and distribute all or any part of it and distribute for commercial purposes.

ইঞ্জিনিয়ার আজিজুল বারী
engineerazizulbari@gmail.com

খানক্বায়ে আমীনিয়া-আসগরিয়া গবেষণা বিভাগ
Ali Centre, Subidbazar, Sylhet, BD

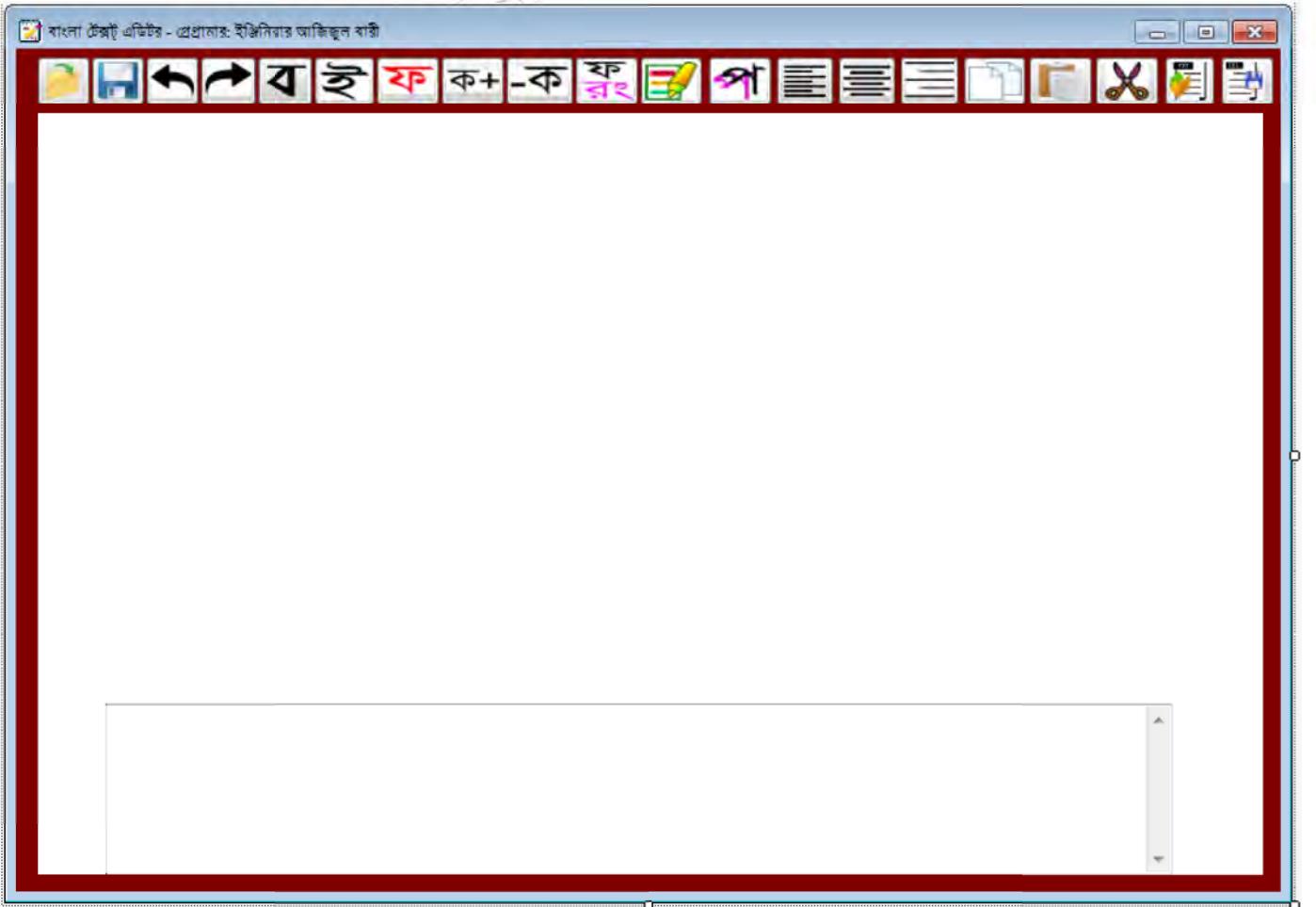
১৪তম পরিচ্ছেদ: *Complete Application Tutorials*

পূর্ণাঙ্গ এ্যাপলিকেশন প্রশিক্ষণ : *Complete Application Tutorials*

Application 4 : Bangla / English Word Processor - বাংলা / ইংরেজি ওয়ার্ড প্রসেসর।

ই্যা, ভিজুয়াল বেসিক.কম দ্বারা ওয়ার্ড প্রসেসর তৈরি করা যায়- আর তা আসলে খুব কঠিন কিছু নয়! অসংখ্য ওয়ার্ড প্রসেসর থাকাসত্ত্বেও আরেকটি কেনো? জবাবে বলা যায়, কেনো নয়? আমরা তো শিখবো- ঠিক না? তাছাড়া বাংলার ওপর গুরুত্ব দিয়ে এটি আমরা তৈরি করতে চাই। সুতরাং আসুন, গড়ে তুলি আমাদের চতুর্থ এ প্রোগ্রামটি- বাংলা-ইংরেজি ওয়ার্ড প্রসেসর।

১. **ইউআই ডিজাইন:** প্রথমেই আমরা ডিজাইন করবো একটি আকর্ষণীয় ইউজার ইন্টারফেইস। নিচের চিত্রে একটি স্যাম্পুল দেখিয়েছি। শিক্ষার্থীরা চাইলে অন্যভাবেও ডিজাইন করতে পারেন। বরং চেষ্টা করবেন, নিজের মনের মতো ডিজাইন করতে। তবে মনে রাখবেন, নিম্নে তালিকাভুক্ত বস্তুগুলো ফর্মে থাকতে হবে।



(ক) একটি উইন্ডোজ ফর্ম:

(১) টেক্সট : “বাংলা টেক্সট এডিটর” [আমি নিজের নামটিও লিখেছি। আপনি আপনার নাম লিখতে পারেন]। (২) অটোস্কেলমোড [AutoScaleMode] = “Font”। (৩) পেছনের রং = Maroon। (৪) ফন্ট = “Times New Roman, 14pt”। (৫) সম্ভব হলো একটি ‘প্রোগ্রাম’ আইকোন। [আমি একটি ও বাকী সব আইকোন নিজে ডিজাইন করে দিয়েছি। সব আইকোন ছবি প্রজেক্টে এ্যামবেড [যুক্ত] করেছি। কিভাবে করা যায়- তা একটু পরই লিখে দেবো। (৬) ফর্মের সাইজ = ৮৭৬, ৬০৮। ব্যস! বাকি সব প্রোগ্রামটি যেভাবে আছে সেভাবেই থাকবে।

১৪তম পরিচ্ছেদ: *Complete Application Tutorials*

পূর্ণাঙ্গ এ্যাপলিকেশন প্রশিক্ষণ : *Complete Application Tutorials*

```

RichTextBox1.ContextMenu = contextMenu1
End Sub
'ফর্মের রিসাইজ ইভেন্ট সাবের কোড। ফর্মের সাইজ পরিবর্তন হলে রিচ টেক্সট বাক্সের সাইজ ও পরিবর্তন হতে হবে।
Private Sub Create_Resize(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Resize
If Me.Size.Width > RichTextBoxSize + 50 Then 'যদি ফর্মের প্রস্থ সাইজ হয় বাক্সের সাইজ + ৫০ পয়েন্ট থেকে বেশি-
RichTextBox1.Width = Me.Size.Width - 100 'তাহলে বাক্সের প্রস্থ সাইজ হবে একশ পয়েন্ট কম।
RichTextBox1.Anchor = AnchorStyles.None 'কন্ট্রোলারের ভেতর কোনো বিশেষ স্টাইলে টেক্সট বাক্স থাকবে না।
RichTextBox1.Anchor = AnchorStyles.Top Or AnchorStyles.Bottom 'শুধু উপর ও নিচের পার্শ্বের নিকটে থাকবে।
'নিচের লাইনটির কাজ হলো ফর্মের সেন্টারে টেক্সট বাক্স থাকার নিশ্চয়তা দেওয়া- সর্বদা।
RichTextBox1.Location = New Point((Me.Size.Width - (Me.Size.Width - 100)) / 2, RichTextBox1.Location.Y)
End If
'নিচের কোড দ্বারা ফর্মের ভেতর টেক্সট বাক্সের অবস্থান, প্রস্থ ও আনকর নিয়ন্ত্রণ করা হয়েছে।
If Me.Size.Width <= RichTextBoxSize + 50 Then
If RichTextBox1.Location.X < 10 Then
RichTextBox1.Location = New Point(5, RichTextBox1.Location.Y)
End If
If RichTextBox1.Width + 25 > Me.Size.Width Then
RichTextBox1.Size = New Point(Me.Size.Width - 10 - RichTextBox1.Location.X, RichTextBox1.Size.Height)
End If
RichTextBox1.Anchor = AnchorStyles.None
RichTextBox1.Anchor = AnchorStyles.Top Or AnchorStyles.Bottom Or AnchorStyles.Left Or AnchorStyles.Right
End If
End Sub
'এটা আনডু বোতামের কোড।
Private Sub UndoBt_Click(sender As Object, e As EventArgs) Handles UndoBt.Click
RichTextBox1.Undo() 'যা এই মাত্র করা হয়েছে তা আনডু করবে।
RichTextBox1.Focus()
End Sub
'এটা রিডু বোতামের কোড।
Private Sub RedoBt_Click(sender As Object, e As EventArgs) Handles RedoBt.Click
RichTextBox1.Redo() 'যা এই মাত্র আনডু হয়েছে তা রিডু করবে।
RichTextBox1.Focus()
End Sub
'এটা বোল্ড তৈরির কোড।
Private Sub BoldBt_Click(sender As Object, e As EventArgs) Handles BoldBt.Click
If RichTextBox1.SelectionFont.Bold = True Then 'যদি বোল্ড থেকে তাকে তাহলে নরম্যাল বানাতে হবে।
If RichTextBox1.SelectionFont.Italic = True Then 'ইটালিক থাকলে সেটা সহ নরম্যাল করতে হবে- যা নিচের কোড লাইনের কাজ।
RichTextBox1.SelectionFont = New Font(Me.RichTextBox1.SelectionFont, FontStyle.Regular +
FontStyle.Italic)
Else 'অন্যথায় ইটালিক ছাড়াই একই কাজ করতে হবে।
RichTextBox1.SelectionFont = New Font(Me.RichTextBox1.SelectionFont, FontStyle.Regular)
End If
Elseif RichTextBox1.SelectionFont.Bold = False Then 'যদি বোল্ড না থাকে তাহলে বোল্ড করতে হবে।
If RichTextBox1.SelectionFont.Italic = True Then 'ইটালিক থাকলে সেটা সহ বোল্ড করতে হবে- যা নিচের কোড লাইনের কাজ।
RichTextBox1.SelectionFont = New Font(Me.RichTextBox1.SelectionFont, FontStyle.Bold + FontStyle.Italic)
Else 'অন্যথায় ইটালিক ছাড়াই একই কাজ করতে হবে।
RichTextBox1.SelectionFont = New Font(Me.RichTextBox1.SelectionFont, FontStyle.Bold)
End If
End If
RichTextBox1.Focus() 'ফোকাস হবে টেক্সট বাক্সে।
End Sub
'এটা হলো ইটালিক বোতামের কোড।
Private Sub ItalicBt_Click(sender As Object, e As EventArgs) Handles ItalicBt.Click
If RichTextBox1.SelectionFont.Italic = True Then 'ইটালিক থাকলে নরম্যাল হবে।
If RichTextBox1.SelectionFont.Bold = True Then 'বোল্ড আছে কি? থাকলে নিচের লাইন নিশ্চিত করবে বোল্ড সহ নরম্যাল করা।

```


১৪তম পরিচ্ছেদ: *Complete Application Tutorials*

পূর্ণাঙ্গ এ্যাপলিকেশন প্রশিক্ষণ : *Complete Application Tutorials*

```

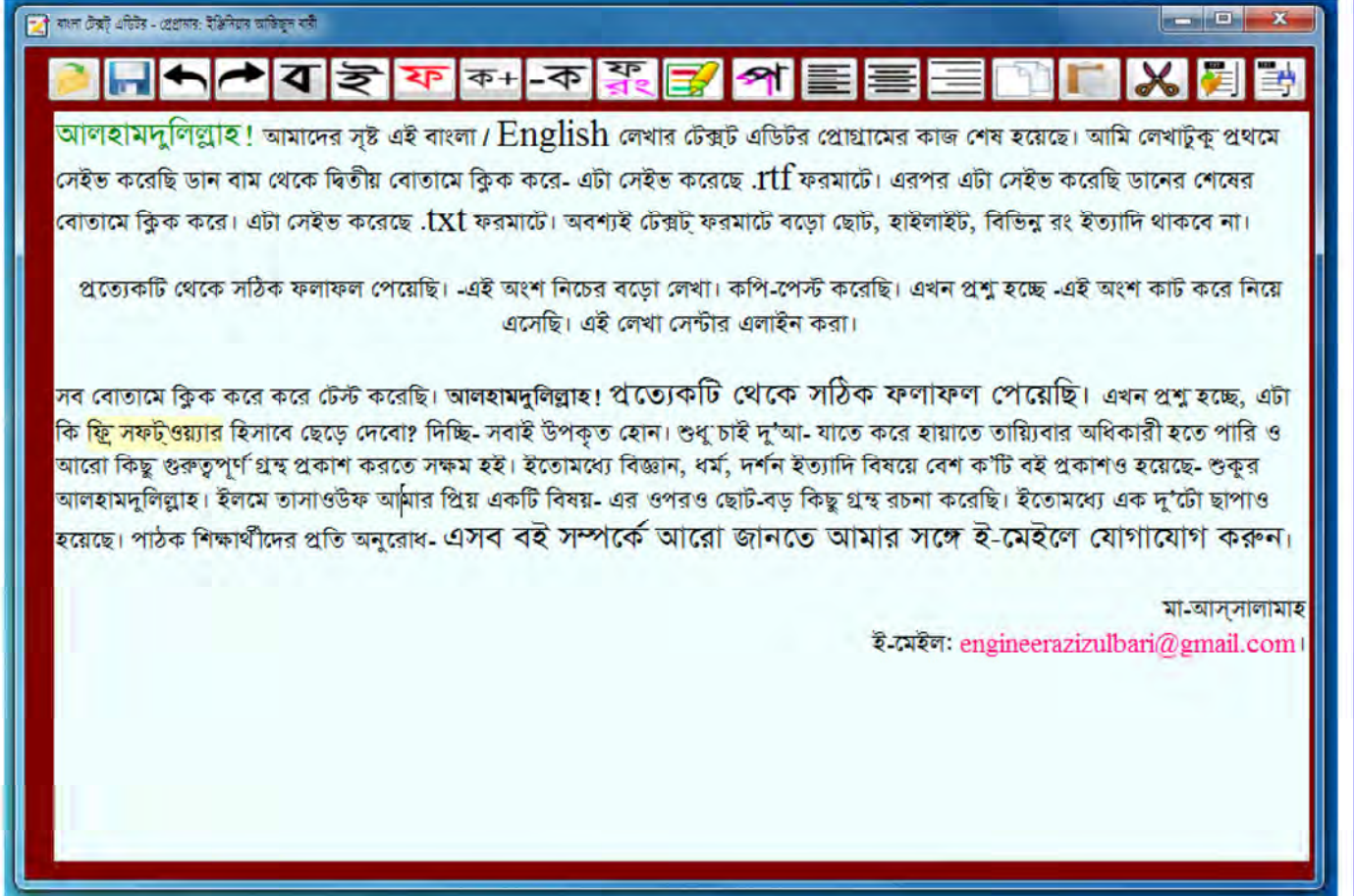
    tooltip1.SetToolTip(BckColBt, s)
End Sub
Private Sub BoldBt_MouseEnter(sender As Object, e As EventArgs) Handles BoldBt.MouseEnter
    Dim s As String = "বল্ড লেখা"
    Dim tooltip1 As System.Windows.Forms.ToolTip = New System.Windows.Forms.ToolTip()
    tooltip1.SetToolTip(BoldBt, s)
End Sub
Private Sub Button1_MouseEnter(sender As Object, e As EventArgs) Handles OpenFileBt.MouseEnter
    Dim s As String = "ফাইল খুলুন"
    Dim tooltip1 As System.Windows.Forms.ToolTip = New System.Windows.Forms.ToolTip()
    tooltip1.SetToolTip(OpenFileBt, s)
End Sub
Private Sub Button2_MouseEnter(sender As Object, e As EventArgs) Handles SaveFileBt.MouseEnter
    Dim s As String = "সংরক্ষণ করুন"
    Dim tooltip1 As System.Windows.Forms.ToolTip = New System.Windows.Forms.ToolTip()
    tooltip1.SetToolTip(SaveFileBt, s)
End Sub
Private Sub CopyBt_MouseEnter(sender As Object, e As EventArgs) Handles CopyBt.MouseEnter
    Dim s As String = "কপি করুন"
    Dim tooltip1 As System.Windows.Forms.ToolTip = New System.Windows.Forms.ToolTip()
    tooltip1.SetToolTip(CopyBt, s)
End Sub
Private Sub CutBt_MouseEnter(sender As Object, e As EventArgs) Handles CutBt.MouseEnter
    Dim s As String = "কেটে দিন"
    Dim tooltip1 As System.Windows.Forms.ToolTip = New System.Windows.Forms.ToolTip()
    tooltip1.SetToolTip(CutBt, s)
End Sub
Private Sub FontBt_MouseEnter(sender As Object, e As EventArgs) Handles FontBt.MouseEnter
    Dim s As String = "ফন্ট পরিবর্তন"
    Dim tooltip1 As System.Windows.Forms.ToolTip = New System.Windows.Forms.ToolTip()
    tooltip1.SetToolTip(FontBt, s)
End Sub
Private Sub FontColBt_MouseEnter(sender As Object, e As EventArgs) Handles FontColBt.MouseEnter
    Dim s As String = "ফন্ট রং"
    Dim tooltip1 As System.Windows.Forms.ToolTip = New System.Windows.Forms.ToolTip()
    tooltip1.SetToolTip(FontColBt, s)
End Sub
Private Sub HighLiteColBt_MouseEnter(sender As Object, e As EventArgs) Handles HighLiteColBt.MouseEnter
    Dim s As String = "হাইলাইট রং"
    Dim tooltip1 As System.Windows.Forms.ToolTip = New System.Windows.Forms.ToolTip()
    tooltip1.SetToolTip(HighLiteColBt, s)
End Sub
Private Sub ItalicBt_MouseEnter(sender As Object, e As EventArgs) Handles ItalicBt.MouseEnter
    Dim s As String = "ইটালিক লেখা"
    Dim tooltip1 As System.Windows.Forms.ToolTip = New System.Windows.Forms.ToolTip()
    tooltip1.SetToolTip(ItalicBt, s)
End Sub
Private Sub PstBt_MouseEnter(sender As Object, e As EventArgs) Handles PstBt.MouseEnter
    Dim s As String = "পেস্ট করুন"
    Dim tooltip1 As System.Windows.Forms.ToolTip = New System.Windows.Forms.ToolTip()
    tooltip1.SetToolTip(PstBt, s)
End Sub
Private Sub RedoBt_MouseEnter(sender As Object, e As EventArgs) Handles RedoBt.MouseEnter
    Dim s As String = "আবার করুন"
    Dim tooltip1 As System.Windows.Forms.ToolTip = New System.Windows.Forms.ToolTip()

```


১৬তম পরিচ্ছেদ: *Complete Application Tutorials*

পূর্ণাঙ্গ এ্যাপলিকেশন প্রশিক্ষণ : *Complete Application Tutorials*

আমরা মোট ৪টি ব্যবহারযোগ্য মোটামুটি জটিল এ্যাপলিকেশন তৈরি করতে সক্ষম হয়েছি। আরো দু'টি প্রোগ্রাম ডেভেলোপ করবো শিক্ষার্থীদের সুবিধার জন্য। তবে এক্ষুণি নয়। পরের পৃষ্ঠাসমূহে ভিজুয়াল বেসিক.নেট -এর একটি শক্তিশালী ডেভেলোপমেন্ট টুল সম্পর্কে কিছুটা আলোকপাত করার প্রয়াস পাচ্ছি। চাক্ষুস্থান শিক্ষার্থীদের মনে হয়তো প্রশ্ন জেগেছে, অনেক গুরুত্বপূর্ণ কম্পিউটার সফটওয়্যার সিস্টেম 'ডাটাবেইজ' সম্পর্কে এই নীরবতা কেনো? এ



প্রশ্নের জবাবেই পরের পুরো অনুচ্ছেদটির অবতারণা। ডাটাবেইজ করার জন্য পর্যাপ্ত টুলস ও ফাংশন অবশ্যই .নেট-এ আছে। আমরা অবশ্যই এডভান্স ডাটাবেইজ প্রোগ্রামিং সম্পর্কে কিছু বলবো না। এটা এই গ্রন্থের স্কোপের বাইরে। তবে বেসিক বিষয়গুলো অবশ্যই জেনে নেবেন। আমরা শেষের দিকে একটি বাংলা ঠিকানা বই ডাটাবেইস তৈরিও করবো। শিক্ষার্থীরা মৌলিক বিষয়গুলো রপ্ত করে নিতে পারলে, ধীরে ধীরে এডভান্স ডাটাবেইজও ডেভেলোপ করতে পারবেন নিশ্চয়ই। কারণ এটুকু পরই দেখতে পাবেন, মাইক্রোসফটের 'Access' ডাটাবেইজের টেবিল কিভাবে অতি

১৫তম পরিচ্ছেদ: Database Programming Made Simple



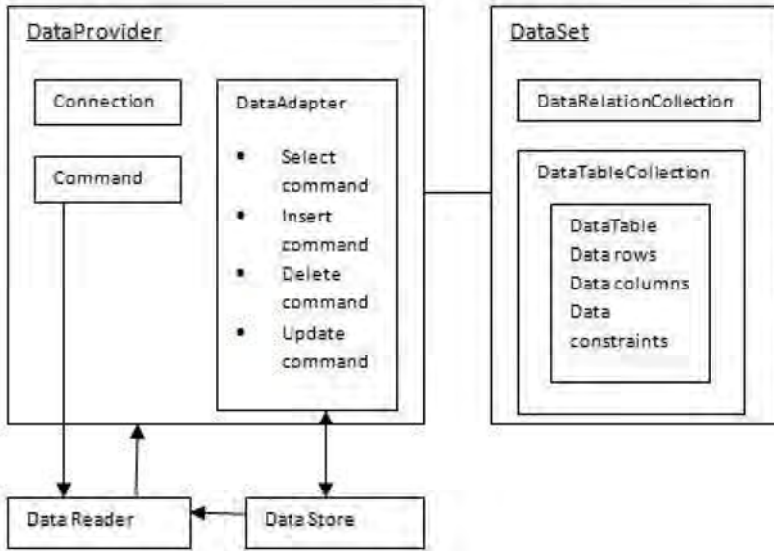
সহজ ডাটাবেইজ প্রোগ্রামিং: Database Programming Made Simple

অনেকেই ভাবেন, ডাটাবেইজ প্রোগ্রামিং বুঝি খুব জটিল-কঠিন। বাস্তবে তা মোটেই নয়। অন্তত, ভিজুয়াল বেসিক.নেট রিলিজ হওয়ার পর থেকে এই ধারণা যে পাল্টে গেছে তা নিশ্চিত করে বলা যায়। এ পরিচ্ছেদ থেকে বিষয়টি অবশ্যই শিক্ষার্থীদের নিকট স্পষ্ট হবে। বাস্তবে সফটওয়্যার ডেভেলপমেন্ট মূলত একটি চিত্তাকর্ষক ক্রিয়ার নাম। তবে যা প্রয়োজন তাহলো, শিক্ষাগ্রহ ও কিছুটা সময়। যাক, এবার আমরা একে একে ডাটাবেইজ প্রোগ্রামিংয়ের ওপর উদাহরণসহ বিস্তারিত আলোচনা করতে চাচ্ছি।

১. এন্টিভএক্স ডাটা অবজেক্ট.নেট [সংক্ষেপে: ADO.Net]

এ্যাপলিকেশন দু'টি উপায়ে সংরক্ষিত ডাটা বা তথ্য নিয়ে কাজ করতে পারে: ১. ব্যবহার-বান্ধব উপায়ে প্রদর্শনের জন্য সংরক্ষিত ডাটা নিয়ে আসা। ২. সংরক্ষিত ডাটাকে প্রয়োজনে পরিবর্তন, পরিবর্ধন, সংযোজন ও মুছে দেওয়া এবং পুনরায় সংরক্ষণ নিশ্চিত করা। মাইক্রোসফট .নেট সৃষ্টিতে এডো.নেট [ADO.Net] নামক একটি মডেল তৈরি করেছে। এটা মূলত .নেট এর ফ্রেমওয়ার্কের একটি অংশ। ফলে .নেট কোনো ডাটাসূত্র যেমন মাইক্রোসফটের Access থেকে ডাটা সহজেই নিয়ে এসে কাজ করতে পারে।

(ক) ADO.Net Object Model - এডো.নেট অবজেক্ট মডেল: এটা মূলত কাঠামোগত একটি প্রসেসের নাম। ডাটাবেইজের সঙ্গে সম্পৃক্ত বিভিন্ন উপাদানে কিভাবে তথ্য চলাফেরা করে তারই একটি উপায়-অবলম্বন। আমরা একে চিত্রের মাধ্যমে তুলে ধরতে পারি। যেমন:



লক্ষ করুন, ডাটা কোনো ডাটাবেইজ বা স্টোরে সংরক্ষিত থাকে। এটা DataProvider দ্বারা বের করে নিয়ে আসা হয়। এই প্রভাইডারের ভেতর বেশ ক'টি অংশ আছে যেগুলোর কাজ হলো ডাটা এ্যাপলিকেশনে নিয়ে আসা যাতেক করে আপডেট করা যায়।

এ্যাপলিকেশনে ডাটা গ্রহণ করে হয় DataSet না হয় DataReader এর মাধ্যমে।

* DataSet তথ্য সংরক্ষণ করে রাখে একটি আলাদা ক্যাশিতে। সেটা এ্যাপলিকেশন নিয়ে আসে।

* DataReader এ্যাপলিকেশনে ডাটা সংযোজন করে শুধুমাত্র পাঠ [read-only] ও অগ্রগামী [forward-only] তথ্য হিসাবে।

(খ) DataProvider - তথ্য সরবরাহকারী : এটা ব্যবহৃত হয় ডাটাবেইজের সঙ্গে সংযুক্তি, কমান্ড এক্সেকিউট, তথ্য আদান-প্রদান, ডাটাসেটে সংরক্ষণ, পড়া ও পরিবর্তন ইত্যাদি কাজের জন্য। ADO.Net -এ ডাটা প্রভাইডারে আছে চারটি অবজেক্ট। যথা:-

(১) Connection - সংযুক্তি: বুঝাই যাচ্ছে এ অংশের কাজ হলো কোনো ডাটা সূত্রের সঙ্গে সংযুক্তি সৃষ্টি করা।

(২) Command - নির্দেশ: কমান্ড হলো কোনো SQL [Structured Query Language] স্টেটমেন্ট কিংবা স্টোরকরা প্রসিডোর। এরই মাধ্যমে ডাটা সূত্রের তথ্যাদি পরিবর্তন, মুছা, সংরক্ষণ ও সংযোজন ইত্যাদি কাজ করা হয়।

(৩) DataReader - তথ্য পাঠক: বুঝাই যাচ্ছে এ অংশের কাজ হলো কোনো ডাটা সূত্র থেকে শুধুমাত্র পাঠ ও

১৫তম পরিচ্ছেদ: Database Programming Made Simple

সহজ ডাটাবেইজ প্রোগ্রামিং: Database Programming Made Simple

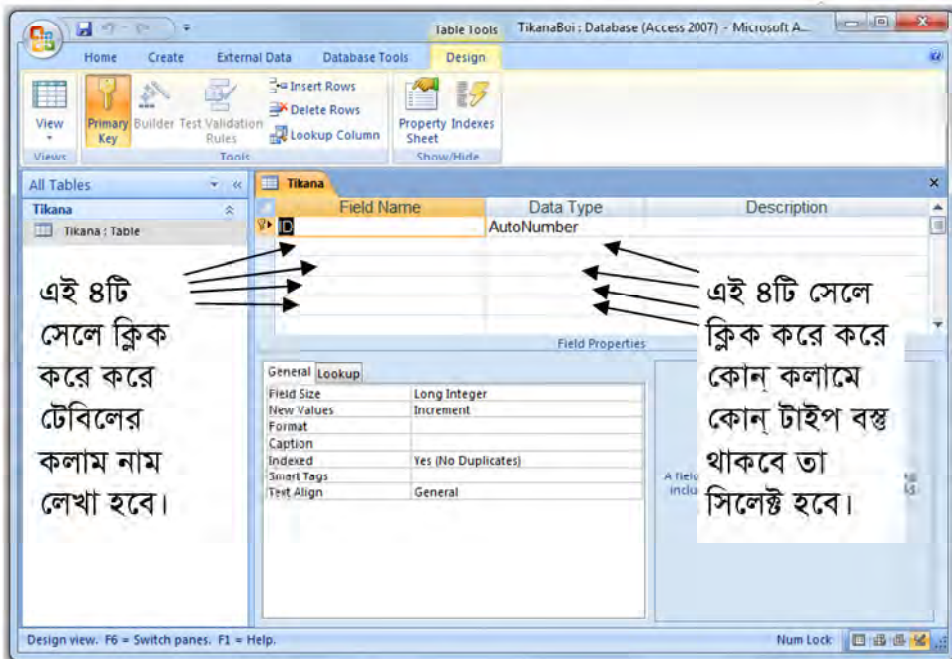
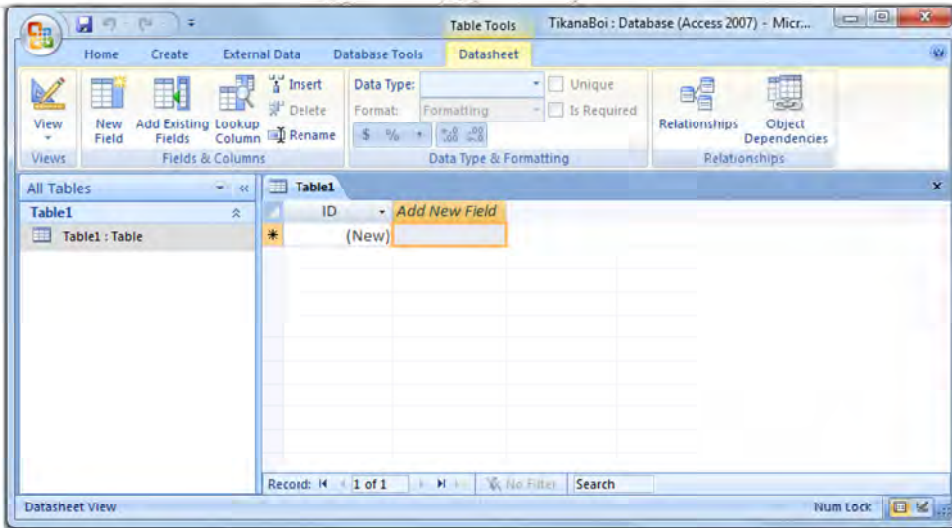
যাক, খুব বেশি টেকনিক্যাল কথা নিয়ে মাথা ঘামাবার প্রয়োজন নেই- উদাহরণের মাধ্যমে যে কোনো বিষয় সম্পর্কে জ্ঞানার্জন একটি উত্তম উপায়। এছাড়া ডাটাবেইজের মতো একটি উচ্চ পর্যায়ের কম্পিউটার প্রোগ্রামিং সম্পর্কে এ গ্রন্থে খুব একটা বেশি বলাও সম্ভব নয়। এ বিষয়ের উপর আলাদাভাবে অনেক গ্রন্থাদি রচিত হয়েছে। আমাদের উদ্দেশ্য হলো, ভিজুয়াল বেসিক.নেট ও ডাটাবেইজ এ দু'য়ের মধ্যে তথ্য আদান-প্রদান সম্পর্কে মৌলিক কিছু জ্ঞানার্জন- এই যা। সুতরাং আসুন, প্রথমে একটি উদাহরণ প্রজেক্ট তৈরি করি।

১. ভিজুয়াল বেসিক.নেট ও ডাটাবেইজ: একটি ঠিকানা বই ডাটাবেইজ প্রোগ্রাম

ঠিকানা বই ডাটাবেইজ প্রোগ্রামের জন্য মাইক্রোসফটের Access ২০০৭ কিংবা এর পরের কোনো ভার্সন আপনার কম্পিউটারে থাকতে হবে। নিম্নের পদক্ষেপগুলো অনুসরণ করে ডাটাবেইজ প্রোগ্রামটি তৈরি করুন।

১. প্রোগ্রাম ফাইলে যেয়ে Access খুলুন। একটি নতুন ডাটাবেইজ তৈরি করুন। এর নামকরণ করুন 'TikanaBoi'। যেখানে ইচ্ছা এটা সেইভ করুন। তবে মনে রাখবেন, পরবর্তিতে এই ফাইলের প্যাথনেইম আমাদের দরকার।

২. প্রথমেই আপনি নিচের চিত্র মুতাবিক জানালাটি দেখতে পাবেন।



এই ৪টি সেলে ক্লিক করে করে টেবিলের কলাম নাম লেখা হবে।

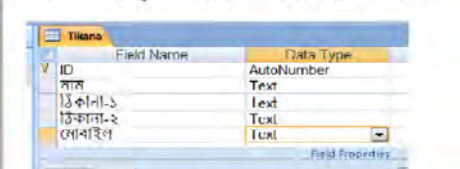
এই ৪টি সেলে ক্লিক করে করে কোন্ কলামে কোন্ টাইপ বস্তু থাকবে তা সিলেক্ট হবে।

৩. বায়ের উপরের কোণে ডানে চিত্রিত আইকনে ক্লিক করুন। যে জানালা আসবে তাতে টেবিলের নাম দিন: 'Tikana'।

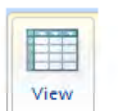
৪. এবার যে জানালাটি আসবে তার একটি চিত্র নিচে দেখানো হয়েছে।

৫. কোন্ কোন্ সেলে আমরা লিখবো তা চিত্রে দেখানো হয়েছে এ্যারো দ্বারা। Field Name কলামের দ্বিতীয় আইটেম হবে: 'নাম', পরেরটি হবে 'ঠিকানা-১', এরপর হবে 'ঠিকানা-২' এবং সবশেষের সেলে হবে 'মোবাইল'।

Data Type কলামের চারটি সেলেই 'Text' শব্দটি থাকবে। দেখুন নিচের চিত্রটি। ব্যস্ ডিজাইন জানালার কাজ



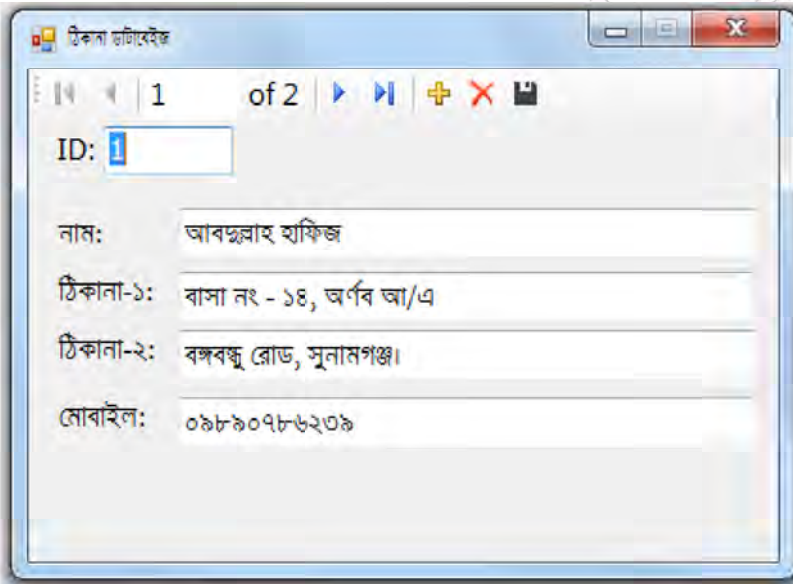
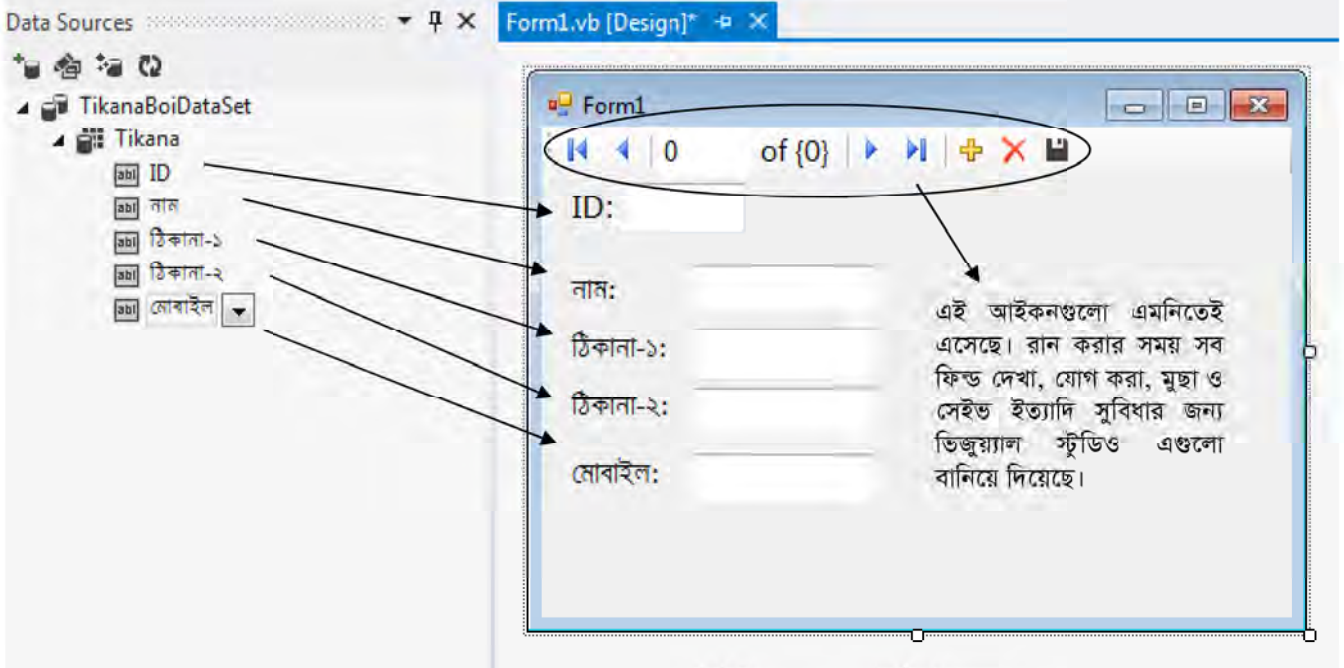
শেষ হলো। এবার ক্লিক করুন উপরের কোণায় স্থাপিত নিচে চিত্রিত আইকনে। একটি ডায়ালগ আসবে সেইভ করার জন্য। এরপর আসবে টেবিল ভিউ, যেখানে আমরা তথ্য লিখতে পারি।



১৫তম পরিচ্ছেদ: *Database Programming Made Simple*

সহজ ডাটাবেইজ প্রোগ্রামিং: *Database Programming Made Simple*

১০. সফলভাবে কানেকশন হওয়ার পর আমরা ফর্মের মধ্যে টেবিলের বিভিন্ন বস্তু টেনে টেনে নিয়ে এসে যেখানে ইচ্ছে বসাতে পারি। নিচের চিত্র দেখুন। আপনি প্রতিটি আইটেম সিলেক্ট করে লেফট-ক্লিক করে টেনে ফর্মের উপর এনে ছেড়ে দেবেন।



লক্ষ করুন, এ পর্যন্ত যাকিছু করলাম, তাতে এক লাইনের কোডিং করতে হয় নি। বাস্তবে ভিজুয়াল স্টুডিও আমাদের জন্য কোডিং করে দিয়েছে। কোডিং নিজে না করলেও যাকিছু উইজার্ডের মাধ্যমে হয়েছে- তা-ই আমাদের ডাটাবেইজ দেখার জন্য যথেষ্ট। এ কথার সত্যতা যাচাই করতে হলে এ পর্যায়ে প্রোগ্রামটি একবার রান করুন। বায়ের চিত্রটি দেখুন। আপনি চাইলে + আইকনে ক্লিক করে নতুন রেকর্ড তৈরি করুন। চারটি ফিল্ড পূরণ করে ডানের সেইভ আইকনে ক্লিক দিন। এরপর প্রোগ্রাম বন্ধ করে আবার খুলুন। দেখবেন, তিনটি রেকর্ড দেখাচ্ছে। শেষের [৩নং] রেকর্ডটি দেখুন।

যাক, কোডিং ছাড়াই একটি শক্তিশালী ডাটাবেইজ তৈরি করতে আমরা সক্ষম হয়েছি। তবে আমাদের ইচ্ছে তো, কোডিং শেখা। সুতরাং

আসুন, কিভাবে কোডিংয়ের মাধ্যমে অনুরূপ ডাটাবেইজ তৈরি করা যায় তা একবার তলিয়ে দেখি।

১৫তম পরিচ্ছেদ: *Database Programming Made Simple*

সহজ ডাটাবেইজ প্রোগ্রামিং: *Database Programming Made Simple*

FullDatabasePath = "C:\Users\htci\Desktop\TikanaBoi.accdb" 'ডেস্কটপে থাকা ডাটাবেইজের পুরো প্যাথ নাম।
 dbSource = "Data Source=" & FullDatabasePath 'dbSource মধ্যে এখন পূর্ণাঙ্গ স্ট্রিং থাকবে।
 'MsgBox(dbSource) 'ডিবাগ করার জন্য এই লাইন। এটি থাকবে না।

এ পর্যন্ত কোড দ্বারা আমরা dbSource স্ট্রিং ভেরিয়েবলে কি হবে তা নিশ্চিত করেছি। অর্থাৎ ডাটা সূত্রের প্যাথ নাম সংরক্ষণ করেছি। কোডের প্রথম লাইনে আমরা যে অবজেক্ট ভেরিয়েবল ডিক্লারার করেছিলাম এখন তাকে কাজে লাগাতে হবে। এজন্য একে দু'টি তথ্য দেওয়া প্রয়োজন: ১. ডাটাবেইজ প্রভাইডার ও ২. ডাটাসূত্র। আমরা উভয়টিই নির্দিষ্ট করে নিয়েছি। অর্থাৎ ডাটাবেইজ প্রভাইডার হলো: "PROVIDER=Microsoft.ACE.OLEDB.12.0;" [dbProvider] এবং ডাটাসূত্র হলো "Data Source = " & FullDatabasePath [dbSource]। আর এ উভয় স্ট্রিং মিলেই তৈরি হয় কানেকশন স্ট্রিংটি। সুতরাং আমরা এবার এ লাইনটি কোডে লিখতে পারি:

```
con.ConnectionString = dbProvider & dbSource
```

অবশেষে আমাদের কানেকশন স্ট্রিং প্রস্তুত। এখন অতি সহজেই আমাদের ডাটাবেইজ খুলতে পারি। এজন্য একটি মাত্র ছোট কোড লাইনের প্রয়োজন:

```
con.Open() -এ কোডটুকু হলো OleDb.OleDbConnection অবজেক্টের একটি ফাংশন।
```

ডাটাবেইজ খোলার পর যাকিছু ইচ্ছে তা সেরে একে আবার ক্লোজ করতে হবে। এটা খুব গুরুত্বপূর্ণ ব্যাপার। ক্লোজ করার জন্যও আমরা আরেকটি সহজ ফাংশন ব্যবহার করে থাকি:

```
con.Close() -ব্যস! ডাটাবেইজ খোলা ও বন্ধ করার কোড শেষ হলো।
```

এ পর্যন্ত কোডে যা হয়েছে তাহলো:

```
Dim con As New OleDb.OleDbConnection 'ভেরিয়েবল con একটি যুক্তকরণ অবজেক্ট হোল্ড করবে।
Dim dbProvider As String 'ভেরিয়েবল dbProvider একটি বিশেষ স্ট্রিং হোল্ড করবে [দেখুন ৬ষ্ঠ লাইনটি]।
Dim dbSource As String 'ভেরিয়েবল dbSource একটি স্ট্রিং হোল্ড করবে।
Dim FullDatabasePath As String 'ডাটাবেইজের পূর্ণাঙ্গ প্যাথনাম।
dbProvider = "Provider=Microsoft.ACE.OLEDB.12.0;" 'সিস্টেমের বিশেষ একটি স্ট্রিং।
FullDatabasePath = "C:\Users\htci\Desktop\TikanaBoi.accdb" 'ডেস্কটপে থাকা ডাটাবেইজের পুরো প্যাথ নাম।
dbSource = "Data Source=" & FullDatabasePath 'dbSource মধ্যে এখন পূর্ণাঙ্গ স্ট্রিং থাকবে।
'MsgBox(dbSource) 'ডিবাগ করার জন্য এই লাইন। এটি থাকবে না।
con.Open() 'ডাটাবেইজ খোলা।
con.Close() 'ডাটাবেইজ বন্ধ করা।
```

আমরা কাজক্ষত গন্তব্যের অর্ধেক পথে এসে গেছি। এবার আরো অগ্রসর হতে হবে- অর্থাৎ, ডাটাবেইজ প্রদর্শনের দরকার এবং বিভিন্ন তথ্য দেখা ও প্রয়োজনে আপডেট করা। পরের পৃষ্ঠাসমূহে আমরা এ সম্পর্কিত কোডগুলো লিখবো। শিক্ষার্থীরা ধৈর্যসহ আমার সাথেই থাকুন এবং স্টাডি করুন।

ডাটাসেট [DataSet] ও ডাটা এডাপটার [Data Adapter]

ডাটা বা তথ্য ডাটাবেজ থেকে কম্পিউটারের মেমোরিতে সংরক্ষণ করার পদ্ধতি হিসাবে আমরা যে সিস্টেম ফাংশন ব্যবহার করি তাকে ADO.Net নামকরণ করেছে DataSet। এটা টেবিলের তথ্যাদির একটি কপি মেমোরিতে হোল্ড করে রাখে। এটা এমন কিছু নয় যা আমাদের ফর্মে লিখে দেখানো যাবে। তবে মেমোরির মধ্যেও এটি কাল্পনিক টেবিল হিসাবে থাকে- অর্থাৎ কলাম ও রো-তে সব তথ্য সংরক্ষিত হয়। একে ডাটাবেইজের ভাষায় ফিল্ড সিস্টেম বলে। তবে প্রশ্ন হচ্ছে DataSet ও Connection Object কি একে অন্যকে দেখতে পারে- বা এমনকি যোগাযোগ করতে সক্ষম?

১৫তম পরিচ্ছেদ: *Database Programming Made Simple*

সহজ ডাটাবেইজ প্রোগ্রামিং: *Database Programming Made Simple*

অনেক ওয়েভপেইজ আছে। যাক, আমরা এখন তিনটি ডিক্লারেশন ও দু'টি লাইন বোতামের কোডে যুক্ত করতে চাই:

```
'=====নতুন ডিক্লারেশন =====
Dim ds As New DataSet 'একটি ডাটাসেট অবজেক্ট হোল্ড করে।
Dim da As OleDb.OleDbDataAdapter 'একটি এডাপটার অবজেক্ট হোল্ড করে।
Dim Sql As String 'একটি সিকুয়েল স্ট্রিং হোল্ড করে।
'=====
'===== নতুন কোড =====
Sql = "SELECT * FROM Tikana" 'সিকুয়েল স্ট্রিং স্টোর করা।
' ডাটা এডাপটারের নিকট সিকুয়েল স্ট্রিং ও কানেকশন অবজেক্ট পাস করে দেওয়া।
da = New OleDb.OleDbDataAdapter(Sql, con)
'=====
```

এখন পর্যন্ত বোতামের কোড হলো:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
Dim con As New OleDb.OleDbConnection 'ভেরিয়েবল con একটি যুক্তকরণ অবজেক্ট হোল্ড করবে।
Dim dbProvider As String 'ভেরিয়েবল dbProvider একটি বিশেষ স্ট্রিং হোল্ড করবে [দেখুন ১০ম লাইনটি]।
Dim dbSource As String 'ভেরিয়েবল dbSource একটি স্ট্রিং হোল্ড করবে।
Dim FullDatabasePath As String 'ডাটাবেইজের পূর্ণাঙ্গ প্যাথনাম।
'=====নতুন ডিক্লারেশন =====
Dim ds As New DataSet 'একটি ডাটাসেট অবজেক্ট হোল্ড করে।
Dim da As OleDb.OleDbDataAdapter 'একটি এডাপটার অবজেক্ট হোল্ড করে।
Dim Sql As String 'একটি সিকুয়েল স্ট্রিং হোল্ড করে।
'=====
dbProvider = "Provider=Microsoft.ACE.OLEDB.12.0;" 'সিস্টেমের বিশেষ একটি স্ট্রিং।
' MsgBox(CurDir)
FullDatabasePath = "C:\Users\htci\Desktop\TikanaBoi.accdb" 'ডাটাবেইজের পুরো প্যাথ নাম।
dbSource = "Data Source=" & FullDatabasePath 'dbSource মধ্যে এখন পূর্ণাঙ্গ স্ট্রিং থাকবে।
'MsgBox(dbSource) 'ডিবাগ করার জন্য এই লাইন। এটি থাকবে না।
con.ConnectionString = dbProvider & dbSource
con.Open()
'===== নতুন কোড =====
Sql = "SELECT * FROM Tikana" 'সিকুয়েল স্ট্রিং স্টোর করা।
' ডাটা এডাপটারের নিকট সিকুয়েল স্ট্রিং ও কানেকশন অবজেক্ট পাস করে দেওয়া।
da = New OleDb.OleDbDataAdapter(Sql, con)
'=====
con.Close()
End Sub
```


১৫তম পরিচ্ছেদ: *Database Programming Made Simple*

সহজ ডাটাবেইজ প্রোগ্রামিং: *Database Programming Made Simple*

```

RecNum = Val(TextBox1.Text) 'ভেরিয়েবলে টেক্সবক্সের সংখ্যা। প্রথমবার এর মান ০।
Dim con As New OleDb.OleDbConnection 'ভেরিয়েবল con একটি যুক্তকরণ অবজেক্ট হোল্ড করবে।
Dim dbProvider As String 'ভেরিয়েবল dbProvider একটি বিশেষ স্ট্রিং হোল্ড করবে।
Dim dbSource As String 'ভেরিয়েবল dbSource একটি স্ট্রিং হোল্ড করবে।
Dim FullDatabasePath As String 'ডাটাবেইজের পূর্ণাঙ্গ প্যাথনাম।
Dim ds As New DataSet 'একটি ডাটাসেট অবজেক্ট হোল্ড করে।
Dim da As OleDb.OleDbDataAdapter 'একটি এডাপটার অবজেক্ট হোল্ড করে।
Dim Sql As String 'একটি সিকুয়েল স্ট্রিং হোল্ড করে।
dbProvider = "Provider=Microsoft.ACE.OLEDB.12.0;" 'সিস্টেমের বিশেষ একটি স্ট্রিং।
FullDatabasePath = "C:\Users\htc\Desktop\LearningDotNet\MyDatabase\TikanaBoi.accdb" 'ডাটাবেইজের পুরো প্যাথ নাম।
dbSource = "Data Source=" & FullDatabasePath 'dbSource এর মধ্যে এখন পূর্ণাঙ্গ স্ট্রিং থাকবে।
con.ConnectionString = dbProvider & dbSource 'কানেকশন স্ট্রিং পূর্ণ হলো।
con.Open() 'ডাটাবেইজ খোলার নির্দেশ।
'ক টি রেকর্ড আছে তা জানা। ভেরিয়েবল cmd একটি সিকুয়েল কমান্ড হোল্ড করবে।
Dim cmd As New OleDb.OleDbCommand("Select COUNT(*) FROM Tikana", con)
NumRec = cmd.ExecuteScalar() 'সিস্টেম ফাংশন দ্বারা রেকর্ড সংখ্যা আনা।
'ফাংশন কল করে রেকর্ড সংখ্যা সঠিক আছে কি না দেখা।
If ValidateEntryNum() = False Then
    con.Close() 'ভুল হলে ডাটাবেইজ বন্ধ করে এক্সিট করা
    Exit Sub
End If
Sql = "SELECT * FROM Tikana" 'সিকুয়েল স্ট্রিং স্টোর করা।
'ডাটা এডাপটারের নিকট সিকুয়েল স্ট্রিং ও কানেকশন অবজেক্ট পাস করে দেওয়া।
da = New OleDb.OleDbDataAdapter(Sql, con)
da.Fill(ds, "TikanaDB") 'দ্বিতীয় প্যারামিটার যে কোনো কিছু হতে পারে।
IDText.Text = ds.Tables("TikanaDB").Rows(RecNum).Item(0) 'আইডি
NameText.Text = ds.Tables("TikanaDB").Rows(RecNum).Item(1) 'নাম
AddressText1.Text = ds.Tables("TikanaDB").Rows(RecNum).Item(2) 'ঠিকানা-১
AddressText2.Text = ds.Tables("TikanaDB").Rows(RecNum).Item(3) 'ঠিকানা-২
MobileText.Text = ds.Tables("TikanaDB").Rows(RecNum).Item(4) 'মোবাইল
EnableNavigationBtn() 'ডাটাবেইজ লোড হয়েছে তাই চারটি ন্যাভিগেশন বোতাম সক্রিয় করতে হবে।
'=====
con.Close() 'ডাটাবেইজ ক্লোজ করা- এটা জরুরী।
End Sub
'এটা প্রথম রেকর্ড দেখার বোতামের সাবরুটিন।
Private Sub btnFirst_Click(sender As Object, e As EventArgs) Handles btnFirst.Click
    TextBox1.Text = "0" 'ট্যাক্সট বক্সে প্রথম রেকর্ড ইনডেক্স সংখ্যা লিখে দেওয়া।
    GetAndFillRecord() 'উপলের রেকর্ড প্রদর্শনের সাবরুটিন কল করা।
End Sub
'এটা সর্বশেষ রেকর্ড দেখার বোতামের সাবরুটিন।
Private Sub btnLast_Click(sender As Object, e As EventArgs) Handles btnLast.Click
    TextBox1.Text = NumRec - 1 'ইনডেক্স শুরু হয় ০ থেকে তাই মোট রেকর্ড থেকে ১ কমাতে হবে।
    GetAndFillRecord() 'রেকর্ড প্রদর্শনের সাবরুটিন কল করা।
End Sub
'এটা পরের রেকর্ড দেখার বোতামের সাবরুটিন।
Private Sub btnNext_Click(sender As Object, e As EventArgs) Handles btnNext.Click
    If ValidateEntryNum() <> False Then 'প্রথমে ভেলিড সংখ্যা আছে কি না দেখতে হবে।
        If TextBox1.Text >= NumRec Then 'ভেলিড সংখ্যা সত্য হলেও, সর্বশেষ রেকর্ডে আসা হয়েছে কিনা দেখতে হবে।
            Beep() 'সর্বশেষ রেকর্ড এটি- সিস্টেম সাউন্ড প্লে করে সাব থেকে বেরিয়ে যাওয়া।
            Exit Sub
        End If
    End If

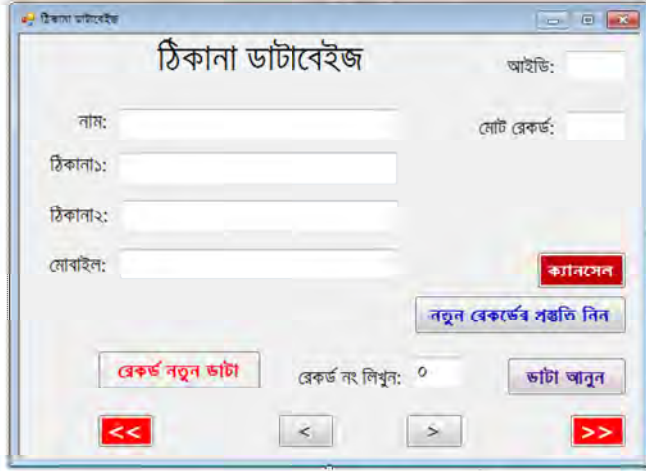
```


১৫তম পরিচ্ছেদ: *Database Programming Made Simple*

সহজ ডাটাবেইজ প্রোগ্রামিং: *Database Programming Made Simple*

ঠিকানা ডাটাবেইজে নতুন রেকর্ড সৃষ্টির কোড

আমরা এ পর্যন্ত যা করেছি তার দ্বারা রেডিমেইড কোনো এক্সেস ডাটাবেইজের তথ্যাদি প্রদর্শন সম্ভব। তবে সময় সময় তো নতুন রেকর্ড প্রবেশ করার প্রয়োজন দাঁড়াতে পারে। সুতরাং আসুন, দেখে নিই কিভাবে কোডের মাধ্যমে তা সম্ভব। প্রথমত, একটি নতুন বোতাম ফর্মে যুক্ত করতে হবে। এর নামকরণ করুন `btnNewRecord`! প্রয়োজনে আমরা ফর্মের সাইজ কিছুটা বড়ো করবো। বোমাতের নামকরণ করুন: `btnNewRecord`।



মনে রাখবেন, বাংলা লিখবেন বিজয়ের ইউনিকোডে। ইউনিকোড আনার শর্টকাট: `Ctrl + Alt + V`। একই শর্টকাট দ্বারা পুনরায় ইংরেজি লিখার কিবোর্ড আনা যায়। যাক, আমরা এবার নতুন বোতামের কোড লিখবো। তবে এর আগে আরো দু'টো বোতাম ফর্মে যুক্ত করা জরুরী।

প্রথমটি হলো একটি 'ক্যানসেল' বোতাম। নতুন ডাটা লেখার বোতামে ক্লিক করার পরও ব্যবহারকারী যাতে ইচ্ছা পরিবর্তন করতে পারেন সে সুযোগ তো থাকতে হবে। দ্বিতীয়টি হলো 'রেকর্ড নতুন ডাটা' নামকরণে একটি বোতাম। নতুন রেকর্ড ফিল্ডে প্রবেশ করার পর এ বোতামে ক্লিক করে ব্যবহারকারী আমাদের ডাটাবেইজ 'ঠিকানাডিবি' -তে তথ্য যুক্ত করবেন-মূলত নতুন একটি তথ্য সারি যোগ হবে। আমরা

বোতামগুলো যুক্ত করে একটি ইউই [User Interface] তৈরি করেছি। উপরের চিত্রটি দেখুন। লক্ষ করুন, 'ক্যানসেল' বোতামের (Name) Property হবে: `btnCancel`। এর Visible Property হবে: `False`। 'রেকর্ড নতুন ডাটা' বোতামের (Name) Property হবে: `btnRecord`। এটার Enabled Property হবে: `False`। এবার আমরা কোডগুলোর প্রতি দৃষ্টি দিতে পারি।

নতুন রেকর্ড যুক্ত করার কোডটি কিছুটা জটিল। সুতরাং খুব সতর্কসহ ব্যাখ্যাটুকু পাঠ করে নেবেন। অবশ্য ইতোমধ্যে যেসব সাবরুটিন সম্পর্কে বলেছি এবং সেগুলো এখনও পরিবর্তিত আছে ওগুলোর ওপর ব্যাখ্যার পুনরাবৃত্তি করবো না। আমরা পুরো প্রোগ্রাম শেষে সমস্ত কোড আবার একসাথে উল্লেখ করবো- অবশ্যই। প্রথমেই যে কাজটি করা জরুরী তাহলো কিছু ডাইমেনশন [Dim] ডিক্লারেশন সর্বত্র প্রাপ্তিযোগ্য করতে হবে। এজন্য অবশ্য ফর্মের ক্লাস ডিক্লারেশনের পরই এসব কোড লিখতে হবে। অর্থাৎ গ্লোবাল ডিক্লারেশন হবে নিম্নরূপ:

'গ্লোবাল ডিক্লারেশন

- `Dim RecNum As Integer = 0` 'এটা হোল্ড করবে টেক্সট বাক্সের সংখ্যাটি যার দ্বারা সঠিক রো এর তথ্য আনা হবে।
- `Dim NumRec As Integer` 'এটা হোল্ড করবে টেবিলের মোট রেকর্ড নাম্বার।
- `Dim KeepingIndex As Integer` 'নতুন রেকর্ড তৈরিতে প্রদর্শিত রেকর্ডের ইনডেক্স নাম্বার সংরক্ষণ
- `Dim ds As New DataSet` 'একটি ডাটাসেট অবজেক্ট হোল্ড করে।
- `Dim da As OleDb.OleDbDataAdapter` 'একটি এডাপটার অবজেক্ট হোল্ড করে।
- `Dim con As New OleDb.OleDbConnection` 'ভেরিয়েবল con একটি যুক্তকরণ অবজেক্ট হোল্ড করবে।
- `Dim dbProvider As String` 'ভেরিয়েবল dbProvider একটি বিশেষ স্ট্রিং হোল্ড করবে।
- `Dim dbSource As String` 'ভেরিয়েবল dbSource একটি স্ট্রিং হোল্ড করবে।
- `Dim FullDatabasePath As String` 'ডাটাবেইজের পূর্ণাঙ্গ প্যাথনাম হোল্ড করবে।
- `Dim Sql As String` 'একটি সিকুয়েল স্ট্রিং হোল্ড করবে।

সবগুলো ভেরিয়েবলের ডিক্লারেশন শেষে কমেন্ট লাইনে বুঝিয়ে দিয়েছি। সুতরাং এখানে আর কিছু যোগ করার নেই- শুধু একটি ছাড়া। এ ব্যাপারটি হলো ভেরিয়েবলগুলোর ডাটা টাইপ। মোট তিনটি ইন্টেজার [অস্তু সংখ্যা], চারটি স্ট্রিং [লেখা] এবং তিনটি অবজেক্ট টাইপ ডিক্লারেশন আছে। এগুলো কিভাবে কাজ করে তা পরবর্তিতে বুঝিয়ে বলছি।

১৫তম পরিচ্ছেদ: *Database Programming Made Simple*

সহজ ডাটাবেইজ প্রোগ্রামিং: *Database Programming Made Simple*

```
৩. 'disables all four navigation btns
Public Sub DisableNavigationBtn()
    btnLast.Enabled = False : btnFirst.Enabled = False
    btnNext.Enabled = False : btnPrev.Enabled = False
    btnCancel.Visible = True
End Sub
```

এই সাবরুটিনের কাজ হলো চারটি ন্যাভিগেশন বোতাম অক্রিয় ও শুধুমাত্র ক্যানসেল বোতামটি প্রদর্শন করা। লক্ষ করুন, কলোন (:) এর ব্যবহার। একই লাইনে একাধিক নির্দেশ দিতে চাইলে এভাবে কলোন ব্যবহার করতে হয়। এই সাবরুটিনটি নতুন রেকর্ডের প্রস্তুতির অংশ।

```
৪. 'deletes five text fields and makes 4 of them editable
Public Sub DeleteTextFromFields()
    NameText.Text = "" : AddressText1.Text = ""
    AddressText2.Text = "" : MobileText.Text = ""
    NameText.ReadOnly = False : AddressText1.ReadOnly = False
    AddressText2.ReadOnly = False : MobileText.ReadOnly = False
    TextBox1.Text = "" 'we dont't want user to edit that field
    Button1.Enabled = False 'disable get record btn
End Sub
```

এটাও নতুন রেকর্ড প্রবেশ করার জন্য প্রস্তুতির অংশ। লক্ষ করুন, ডিজাইন টাইমে আমরা চারটি তথ্যফিল্ড অক্রিয় করে রেখেছিলাম। কারণ, আমরা চাইনি ব্যবহারকারী অসতর্কতাহেতু কোনো রেকর্ডে পরির্তন আনেন। তবে এবার তো নতুন রেকর্ড লিখা হবে তাই প্রথমে সবগুলো মুছে ফেলার পর এগুলো সক্রিয় করেছি। কিন্তু TextBox1 -এ কিছু লেখা আমরা চাই না। তাই একে মুছার পরও অক্রিয় রেখেছি। সব শেষের লাইন দ্বারা 'ডাটা আনুন' বোতামটি অক্রিয় করা হয়েছে।

```
৫. 'this sub locks all 4 flds
Public Sub MakeFourFldReadOnly()
    NameText.ReadOnly = True : AddressText1.ReadOnly = True
    AddressText2.ReadOnly = True : MobileText.ReadOnly = True
End Sub
```

রেকর্ড চুকানো কিংবা ক্যানসেল হওয়ার পর চারটি তথ্যফিল্ডকে পুনরায় অক্রিয় করতে হবে- যাতেকারে ব্যবহারকারী আর কিছু লিখতে না পারেন। এ কাজটিই আজ্ঞাম দেয় উপরোক্ত সাবটি।

```
৬. 'returns true if all 4 flds contains text, ready for record validation
Function isAllRecEntered()
    If NameText.Text <> "" And AddressText1.Text <> ""
        And AddressText2.Text <> "" And MobileText.Text <> "" Then
        Return True
    End If
    Return False
End Function
```

ইতোমধ্যে এ ফাংশনটির কথা উল্লেখ করেছি। এর কাজ হলো টেক্সট চারটিতে লেখা আছে কি না সে বুলিয়ান তথ্য রিটার্ন করা।

```
৭. 'btnCancel event code, simply show the last record, disable itself, enable get rec btn
Private Sub btnCancel_Click(sender As Object, e As EventArgs) Handles btnCancel.Click
    TextBox1.Text = KeepingIndex 'এটার মধ্যে আগে প্রদর্শিত রেকর্ড ইনডেক্স আছে।
    GetAndFillRecord() 'রেকর্ড প্রদর্শনের সাবরুটিন কল করা।
    MakeFourFldReadOnly() 'ফিল্ড চারটে রীড-অনলি বানানো।
    btnCancel.Visible = False : Button1.Enabled = True 'enable get record btn
    btnRecord.Enabled = False : btnNewRecord.Enabled = True
End Sub
```


১৫তম পরিচ্ছেদ: *Database Programming Made Simple*

সহজ ডাটাবেইজ প্রোগ্রামিং: *Database Programming Made Simple*

আগের পৃষ্ঠার ৭ নাম্বারে প্রদর্শিত 'ক্যানসেল' বোতামের সাবরুটিন দ্বারা ইতোমধ্যে যে রেকর্ডটি টেক্স ফিল্ডগুলোতে ছিলো তা পুনরায় দৃশ্যমান করা হয়েছে। এটা প্রথমে ইনডেক্স সংখ্যা টেক্সবাক্স-১ এ লিখে দেয়। এরপর কল করে একে একে ইতোমধ্যে আলোচিত [GetAndFillRecord() এবং MakeFourFldReadOnly() নামক] দু'টি সাবরুটিন। প্রথমটি তো আমাদের মূল সাব- যার কাজ রেকর্ড প্রদর্শন করা আর দ্বিতীয়টির কাজ হলো চারটি ফিল্ডকে লেখার জন্য অক্রিয় করে দেওয়া।

```
৮. 'record Button
Private Sub btnRecord_Click(sender As Object, e As EventArgs) Handles btnRecord.Click
    Dim MsgStr As Integer = MsgBox("সতর্কবার্তা! সবকিছু সঠিক মতো লিখেছেন তো?", MsgBoxStyle.YesNo)
    'MsgBox(MsgStr)
    If MsgStr = 7 Then
        Exit Sub
    End If
    AddRows()
End Sub
```

এই btnRecord ['রেকর্ড নতুন ডাটা'] বোতামের ক্লিক ইভেন্ট কল করে সাবরুটিন 'AddRows()'। এটা ডাকার পূর্বেও একটি সতর্কবার্তা ডায়ালগ দেখানোর ব্যবস্থা করেছি। কারণ ব্যবহারকারী ভুল করতে পারেন তথ্য প্রবেশে- কিংবা তিনি নতুন রেকর্ড এ মুমূর্তে ঢুকানোর ক্রিয়া ক্যানসেলও করার ইচ্ছা রাখতে পারেন। তাকে এই সুযোগটুকু তো দিতে হবে না? প্রথম চার লাইন এ সুযোগ দিয়েছে।

```
৯. Private Sub AddRows()
    If isAllRecEntered() = False Then
        MsgBox("সব রেকর্ড সঠিকভাবে ঢুকিয়েছেন কি না দেখুন! তথ্য রেকর্ড হয় নি।")
        Exit Sub
    End If
    EntryToRaw() 'রেকর্ড ঢুকানোর সাবরুটিন কল করা।
    GetAndFillRecord() 'রেকর্ড প্রদর্শনের সাবরুটিন কল করা।
    MakeFourFldReadOnly() 'ফিল্ড চারটে রীড-অনলি বানানো।
    btnCancel.Visible = False
    Button1.Enabled = True 'enable get record btn
    btnRecord.Enabled = False
    btnNewRecord.Enabled = True
End Sub
```

এই সাবরুটিনইটিই হলো রেকর্ড টেবিলে যোগ করার পূর্ব প্রস্তুতির সর্বশেষ ধাপ। EntryToRaw() নামক সর্বশেষ সাবরুটিন কলের পূর্বে একটি টেস্ট করা হয়। সকল ফিল্ডে টেক্সট প্রবেশ করা হয়েছে কিনা তা জানতে হবে। যদি সকল বাক্সে টেক্সট না থাকে, তাহলে মেসেজ বাক্সের মাধ্যমে সতর্কবাণী প্রদর্শন করে বেরিয়ে যেতে হবে সাবরুটিন থেকে। এরপর [সবটি ঠিকঠাক থাকলে] প্রথমে EntryToRaw() সাবরুটিন কল করে নতুন রেকর্ড তৈরি করা হয়। এরপর রেকর্ড প্রদর্শনের মূল সাবরুটিন কল করে আগের প্রদর্শিত তথ্যাদি টেক্সট বাক্সগুলোতে পুনরায় দেখানো হচ্ছে। এর পরের সাবরুটিন কল করে সকল টেক্স বাক্স রিড-অনলি করতে হবে। সবশেষে তিনটি বোতামের ক্রিয়া পরিবর্তন করা জরুরী: ক্যানসেল বোতাম গোপন করা; রেকর্ড আনার বোতাম সক্রিয় করা; রেকর্ড করার বোতাম নিষ্ক্রিয় করা; এবং নতুন রেকর্ডের প্রস্তুতি-বোতাম আবার সক্রিয় করা।

এ হচ্ছে সর্বশেষ ডাটা এনট্রি সাবের পূর্ব পর্যন্ত সাবগুলোর ক্রিয়ার বর্ণনা। আমরা এবার পরের পৃষ্ঠায় এক্সেস টেবিলে তথ্যাদি রেকর্ড করার ক্রিয়াটির পূর্ণাঙ্গ ব্যাখ্যা তুলে ধরবো।

১৫তম পরিচ্ছেদ: *Database Programming Made Simple*

সহজ ডাটাবেইজ প্রোগ্রামিং: *Database Programming Made Simple*

'Code for data entry into access table.

```
Private Sub EntryToRaw()
    Dim cb As New OleDb.OleDbCommandBuilder(da) 'commandBuilder Object creation
    Dim dsNewRow As DataRow 'new data row creation variable
    'notice here, ID column as number will be added automatically [AutoNumber]
    dsNewRow = ds.Tables("TikanaDB").NewRow() 'adding a row into TikanaDB
    dsNewRow.Item(1) = NameText.Text 'name text into 2nd column cell
    dsNewRow.Item(2) = AddressText1.Text 'Address text into 3rd column cell
    dsNewRow.Item(3) = AddressText2.Text 'address text2 into 4th column cell
    dsNewRow.Item(4) = MobileText.Text 'mobile number [as text] into 5th column text
    ds.Tables("TikanaDB").Rows.Add(dsNewRow) 'actual row of data in dataset
    da.Update(ds, "TikanaDB") 'the row of data in ds goes to the created row
    con.Close() 'now close the connection
    MsgBox("ডাটাবেইজে নতুন রেকর্ড সফলভাবে প্রবেশ করা হয়েছে!") 'show success message
End Sub
```

সাবের প্রথম লাইন: কমান্ড বিন্ডার সিস্টেম অবজেক্টের নতুন ইন্সট্যান্ট ডিক্লারেশন (আমাদের নামকরণ: cb)। এতে একটি প্যারামিটার আছে- তাহলো ডাটা এডাপটার [আগেই ডিক্লার করা এই এডাপটারের আমাদের কর্তৃক নামকরণ: da]।

দ্বিতীয় লাইন: একটি ডাটারো টাইপ ডিক্লারেশন [টেবিলে নতুন তথ্যলাইন সৃষ্টির জন্য এটি, আমাদের নামকরণ: dsNewRow]

তৃতীয় লাইন: উক্ত ডাটাটাইপ ভেরিয়েবলে ঠিকানা ডাটাবেইজের টেবিলে একটি নতুন রো [বা তথ্যলাইন] সৃষ্টি করে dsNewRow -তে সংরক্ষণের নির্দেশ। লক্ষ করুন, ds.Tables নির্দেশনার একটি প্যারামিটার হলো আমাদের TikanaDB। এখানে ds হলো পূর্বেই সৃষ্ট আমাদের DataSet। সবশেষে .NewRow() দ্বারা পয়েন্ট করা হচ্ছে নতুন রো সৃষ্টির ফাংশনের দিকে। শিক্ষার্থীদের প্রতি অনুরোধ, এসব সিস্টেম নির্দেশনা কিভাবে কাজ করে, সে সম্পর্কে কোনো ভাবনার প্রয়োজন নেই। এতো গভীরে যাওয়া আমাদের কাজ নয়- বরং ওসব কাজ সফটওয়্যার বিজ্ঞানীদের। অর্থাৎ মাইক্রোসফটের ইঞ্জিনিয়ারদের। আমরা তাদের আবিষ্কার ও উদ্ভাবনগুলো কাজে লাগাবো কিভাবে- সেটাই জানতে হবে।

চতুর্থ থেকে সপ্তম লাইন: এই লাইনগুলো দ্বারা ব্যবহারকারী কর্তৃক চারটি ফিল্ডে প্রবেশকৃত টেক্সট ইনডেক্স নির্দেশনার মাধ্যমে ডাটাসেটে সংরক্ষণ করেছি। লক্ষ করুন, আইডি সংখ্যা সংরক্ষণের প্রয়োজন নেই। এটা এক্সেস নিজেই বানাবে। কারণ আমরা এই সেলকে AutoNumber ডটাটাইপ হিসাবে তৈরি করেছি। এ পর্যন্ত সকল নির্দেশনা ডাটাসেট এর মধ্যে ds নামকরণে মেমোরিতে সংরক্ষণ করেছি। সবকিছু প্রস্তুত হয়ে গেছে। এবার সত্যিকার অর্থে টেবিলের মধ্যে একটি তথ্যলাইন যোগ করার নির্দেশ দিতে হবে।

অষ্টম লাইন: এই লাইনটি কিভাবে কাজ করে তা একটু বুঝার দরকার। মনে রাখবেন, প্রতিটি ডট [.] হলো পয়েন্টার ইঙ্গিত। প্রথমে ডাটাসেট [ds] পয়েন্ট করছে আমাদের ঠিকানা টেবিলের দিকে (.TikanaDB)। এরপর পয়েন্ট করছে রো বা তথ্যলাইনের দিকে [.Raw]। সবশেষে পয়েন্ট করছে নতুন রো সৃষ্টির দিকে [.Add(dsNewRow)]। এ লাইনের কোড রান হওয়ার পর আমাদের এক্সেস টেবিলে একটি নতুন তথ্যলাইন বাস্তবে যোগ হবে। এতে শুধুমাত্র আইডি ফিল্ডে একটি ইউনিক [আলাদা] আইডি সংখ্যা থাকবে। বাকী চারটি সেল এখনও শূন্য। পরের লাইনের নির্দেশ দ্বারা এগুলো পূরণ করা হবে।

নবম লাইন: ds.Update একটি ফাংশন। এতে আছে দু'টি প্যারামিটার (ds, "TikanaDB")। প্রথমটি আমাদের সৃষ্ট ডেটাসেট যেখানে সংরক্ষিত আছে চারটি টেক্সট বাক্সের তথ্যাদি। দ্বিতীয় প্যারামিটার হলো আমাদের এক্সেস ডাটাবেইজের নাম। ব্যস! তথ্যাদি সেলের মধ্যে চলে যাবে।

দশম লাইন: কাজ শেষ। এবার কানেকশন বন্ধ করার পালা। এই লাইনে তা-ই নির্দেশ করা হয়েছে।

শেষ লাইন: ব্যবহারকারীকে তথ্য দেওয়া হচ্ছে- নতুন তথ্যলাইন তৈরি হয়েছে সফলভাবে। এখন প্রোগ্রাম বন্ধ করে পুনরায় চালালে নতুন রেকর্ডও দেখাবে।

১৫তম পরিচ্ছেদ: *Database Programming Made Simple*

সহজ ডাটাবেইজ প্রোগ্রামিং: *Database Programming Made Simple*

'row delete codes

```
Sub DeleteRow()
    If NumRec = 0 Then 'incase there aren't any row of data
        MsgBox("টেবিলে তো আর কোনো রেকর্ড নেই!")
    Exit Sub
End If
Dim cb As New OleDb.OleDbCommandBuilder(da) 'commandBuilder Object creation
ds.Tables("TikanaDB").Rows(RecNum).Delete() 'actual delete row command
da.Update(ds, "TikanaDB") 'update database
TextBox1.Text = RecNum - 1
MsgBox("এই রেকর্ড মুছে গেছে।") 'show success message
con.Close() 'close connection
End Sub
```

প্রথমে টেস্ট। যা কাটা হবে তা আছে কি? না থাকলে মেসেজ প্রদর্শন করে সাব থেকে বের হতে হবে। মনে রাখবেন, NumRec একটি গ্লোবাল ভেরিয়েবল, যার মধ্যে সংরক্ষিত আছে মোট রেকর্ড সংখ্যা- এটা ইনডেক্স সংখ্যা থেকে ১ বেশি। রেকর্ড থাকলে আমরা পরের লাইনে যেয়ে প্রথমে cb নামক একটি কমান্ড বিল্ডার অবজেক্ট ডিক্লয়ার করেছি। এর প্যারামিটার হলো ডাটাএডাপটার (da)। পরের লাইনে আমাদের ডাটাসেট [ds] এর মাধ্যমে পয়েন্ট করছি ঠিকানা ডাটাবেইজের টেবিল থেকে একটি তথ্যলাইন মুছে ফেলতে। Rows(RecNum) নির্দেশ করছে কোন্টি মুছা হবে। মনে রাখা দরকার, বর্তমানে প্রদর্শিত রেকর্ডের ইনডেক্স সংখ্যা হলো RecNum। সুতরাং আমরা এটিই ডিলিট করার কথা বলেছি। ঠিকানা ডাটাবেইজ এরপর আপডেট করতে হবে। বাস্তবে এই আপডেট নির্দেশই এক্সেসের ফিজিক্যাল ফাইলে রবদল নিশ্চিত করে। পরের লাইনের কাজ হলো ইনডেক্স সংখ্যা ১ কমানো- কারণ এখন একটি ইনডেক্স কমে গেছে। এরপর একটি মেসেজ বাক্সে রেকর্ড ডিলিটের বার্তাটি ব্যবহারকারীকে দেওয়া হয়েছে। আর সবশেষে কানেকশন বন্ধ করার নির্দেশ এসেছে।

এবার 'এই রেকর্ড এডিট করুন' ও 'সেইভ করুন' বোতামদ্বয়ের কোড নিয়ে আলোচনার প্রয়াস পাচ্ছি। আর এরই মাধ্যমে ডাটাবেইজ প্রোগ্রামিংয়ের ওপর আলোচিত এ পরিচ্ছেদের ইতি টানবো। শিক্ষার্থীরা এ থেকে যা শিখেছেন, সাধারণ ডাটাবেইজ প্রোগ্রামিংয়ে জন্য যথেষ্ট হবে বলে মনে করছি। তবে এর ওপর আরো গভীর জ্ঞানার্জনের জন্য ভিজুয়াল বেসিক.নেট দিয়ে ডাটাবেইজ সৃষ্টির উন্নতমানের বই-পুস্তক পাঠ ও ইন্টারনেটে প্রাপ্ত তথ্যাদি অধ্যয়ন করতে পারেন।

'editing btn code

```
Private Sub btnEdit_Click(sender As Object, e As EventArgs) Handles btnEdit.Click
    'confirmation test
    Dim ResP As Integer = MsgBox("এই রেকর্ডে রদবদল করবেন?", MsgBoxStyle.YesNo)
    If ResP = 7 Then
        Exit Sub
    End If
    EnableFourFlds() 'enable fields for editing
    btnUpdate.Visible = True 'show the update button
    Button1.Enabled = False
    btnDelete.Enabled = False
    btnNewRecord.Enabled = False
End Sub
```

আমাদেরকে আরোও একটি টেস্ট করতে হবে। কারণ, ব্যবহারকারী এই বোতামে ভুলেও ক্লিক করতে পারেন। টেস্টে পাশ হওয়ার পর প্রথমে ফিল্ডগুলো সক্রিয় করতে একটি সাবরুটিন ডাক দিয়েছি। এরপর সেইভ করুন বোতামটি দৃশ্যমান করেছি। এছাড়া 'নতুন রেকর্ডের প্রস্তুতি নিন', 'ডাটা আনুন' এবং 'এই রেকর্ড ডিলিট করুন' -এ তিনটি বোতামকে নিষ্ক্রিয় করতে হয়েছে। এগুলোর কোনো ফাংশন নেই এডিটের পূর্ব পর্যন্ত।

১৭তম পরিচ্ছেদ: *Database Programming Made Simple*

সহজ ডাটাবেইজ প্রোগ্রামিং: *Database Programming Made Simple*

```
Private Sub btnUpdate_Click(sender As Object, e As EventArgs) Handles btnUpdate.Click
    Dim ResP As Integer = MsgBox("আপনি কি সব রেকর্ড সঠিকভাবে প্রবেশ করেছেন?", MsgBoxStyle.YesNo)
    If ResP = 7 Then
        Exit Sub
    End If
    UpdateRecord() 'call update sub
    MakeFourFldReadOnly() 'call sub to lock flds
    GetAndFillRecord() 'fill # of records % of records & total
    btnUpdate.Visible = False
    Button1.Enabled = True
    btnDelete.Enabled = True
    btnNewRecord.Enabled = True
End Sub
```

এখানেও একটি টেস্ট আছে! ব্যবহারকারী সঠিকভাবে সবকিছু এডিট করেছেন কি না জেনে নিতে হবে। টেস্টের ফলাফল ইতিবাচক হলে আমরা কল করতে পারি UpdateRecord() নামক সর্বশেষ সাবরুটিনটি। আপডেট হওয়ার পর চারটি ফিল্ড রিড-অনলি বানানো ও একটি বোতাম নিষ্ক্রিয় ও তিনটি সক্রিয় করা দরকার। শেষের একটি সাবরুটিন ও চারটি লাইন এ কাজগুলো সারবে।

এবার দেখে নেই সর্বশেষ কোড ব্লকটি। এটা একটি সাবরুটিন যার কাজ হলো এডিট করা লেখাগুলো স্ব স্ব স্থানে নিয়ে যাওয়া- তথা তথ্যাদি আপডেট করা।

```
Private Sub UpdateRecord()
    Dim cb As New OleDb.OleDbCommandBuilder(da) 'commandBuilder Object creation
    ds.Tables("TikanaDB").Rows(RecNum).Item(1) = NameText.Text
    ds.Tables("TikanaDB").Rows(RecNum).Item(2) = AddressText1.Text
    ds.Tables("TikanaDB").Rows(RecNum).Item(3) = AddressText2.Text
    ds.Tables("TikanaDB").Rows(RecNum).Item(4) = MobileText.Text
    da.Update(ds, "TikanaDB") 'update database
    MsgBox("রেকর্ড রদবদল হয়েছে।") 'show success
    con.Close() 'close connection
End Sub
```

সেইভ করুন [আপডেট করার] বোতামটির কোড বাস্তবে খুব এটা কঠিন কিছু না। প্রথমে -এর নতুন একটি ইন্সট্যান্স OleDb.OleDbCommandBuilder সৃষ্টি করতে হবে। এর প্যারামিটার হবে আমাদের সৃষ্ট ডাটাএডাপটার [da]। এরপর চারটি লাইনে ডাটাসেটে নিয়ে যেতে হবে নির্দিষ্ট ইনডেক্সের [ds] চারটি আইটেম [১,২,৩ ও ৪]। সব প্রস্তুত। এবার da.Update(ds, "TikanaDB") লাইনটির মাধ্যমে এক্সেস ডাটাবেইজের টেবিলটি আপডেট করা হয়েছে। এরপর মেসেজ বাক্সের মাধ্যমে সফলতা জানানো ও কানেকশন ক্লোজ করা হয়েছে। আলহামদুলিল্লাহ! ঠিকানা ডাটাবেইজের কাজ পূর্ণাঙ্গভাবে শেষ হলো। শিক্ষার্থীদের প্রতি আবারো অনুরোধ, শুরু থেকে শেষ পর্যন্ত কোডের ব্যাখ্যা প্রয়োজনে একাধিকবার পাঠ করে বুঝার চেষ্টা করবেন। আর অবশ্যই, নিজে এই ডাটাবেইজ প্রোগ্রামটি সৃষ্টির প্রাকটিস করবেন।

শেষ হলো ডাটাবেইজ প্রোগ্রামিংয়ের ওপর রচিত এ পরিচ্ছেদটি। এবার আরো এক দু'টো স্টেপ বাই স্টেপ মডেল প্রোগ্রাম বানানোর প্রশিক্ষণ উপস্থাপনের চেষ্টা করবো। আসুন, আপনাদের সহযোগিতায় গড়ে তুলি ওসব ব্যবহারেপযোগী এ্যাপলিকেশন।

১৬তম পরিচ্ছেদ: *Few More Sample Applications*



আরো কয়েকটি নমুনা এ্যাপলিকেশন: *Few More Sample Applications*

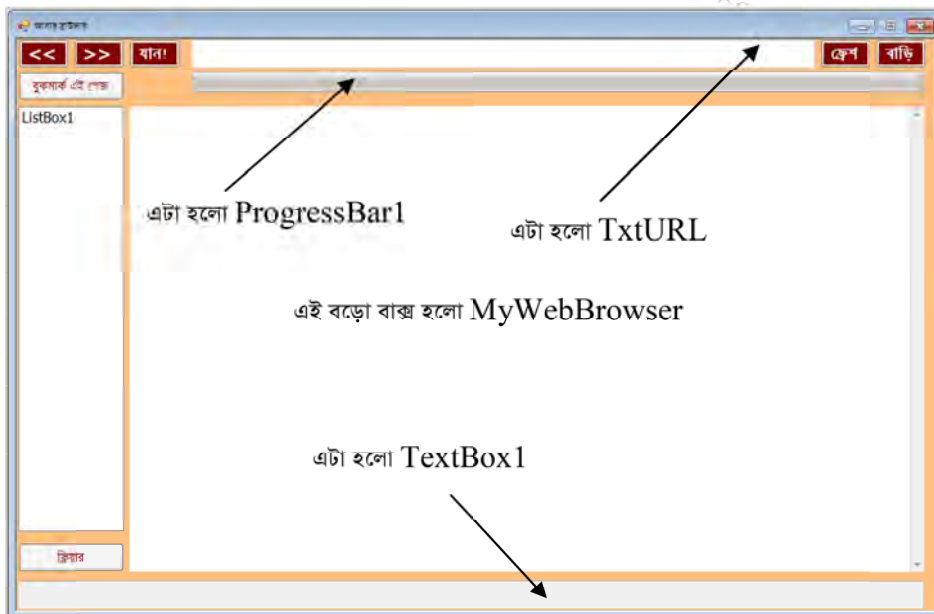
এ পরিচ্ছেদে আরো কয়েকটি নমুনা এ্যাপলিকেশন পদে পদে আমরা গড়ে তুলবো। শিক্ষার্থীরা প্রতিটি পদক্ষেপ সঠিকভাবে অনুসরণ করবেন, যেভাবে ইতোমধ্যে এ গ্রন্থে আলোচিত সকল প্রোগ্রাম তৈরির সময় করেছেন। পরিচ্ছেদ ১৭ শেষ হওয়ার পর ভেবেছিলাম ভিজুয়াল বেসিক.নেট এর উপর রচিত এই বইটির ইতি টানবো। কিন্তু শেষ পর্যন্ত সিদ্ধান্ত পরিবর্তন করলাম এ কারণে যে, কম্পিউটার সফটওয়্যার সৃষ্টির মূলে প্রাকটিস একটি গুরুত্বপূর্ণ ব্যাপার। শিক্ষার্থীরা যতো বেশি প্রাকটিস করবেন ততো বেশি প্রোগ্রামিংয়ের বিভিন্ন কনসেপ্ট সম্পর্কে অভিজ্ঞতা অর্জন করবেন। হতে পারে, এ গ্রন্থের পাঠকদের মধ্যে ভবিষ্যতে দেশের একজন উজ্জ্বল তারকাও কেউ রূপান্তরিত হবেন। আমি অবশ্যই এ আশা করবো। যাক, আমরা একে একে এই নমুনা এ্যাপলিকেশনগুলোর ওপর বিস্তারিত আলোচনা শুরু করছি, এখনই।

১. একটি ব্যক্তিগত ওয়েভ ব্রাউজার: **A Personal Web Browser**

হ্যাঁ, ভিজুয়াল বেসিক.নেট ও ভিজুয়াল স্টুডিও আপনাকে ওয়েভ ব্রাউজার এ্যাপলিকেশনের মতো জটিল প্রোগ্রাম সৃষ্টির সুযোগ তৈরি করে দিয়েছে। আর তা এতোই সহজ! একটু পরই শিক্ষার্থীরা বুঝতে পারবেন। কয়েকটি বিশেষ পদক্ষেপ ও অতি অল্প কিছু কোডিং দ্বারাই তৈরি হবে আপনার নিজস্ব স্বাধীন ওয়েভ ব্রাউজার এ্যাপলিকেশন।

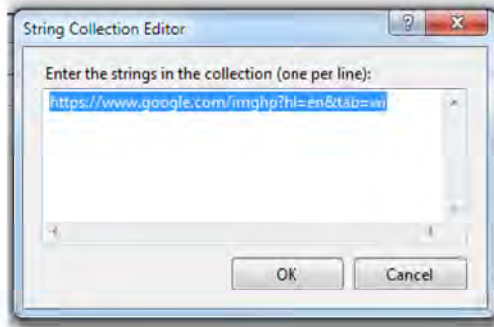
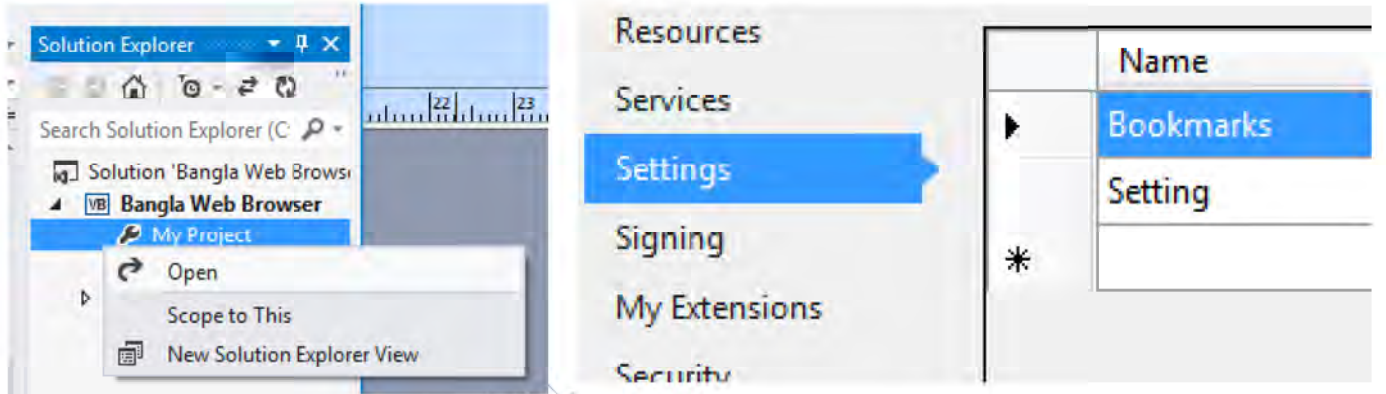
ক. ডিজাইন: নিচে একটি চিত্র দিয়েছি। আপনার ওয়েভ ব্রাউজার এরূপই হবে। এতে আছে ' টি বোতাম, ২টি টেক্সট বাক্স, ১টি লিস্ট বাক্স, ১টি প্রগ্রেস বার ও ১টি ওয়েভ ব্রাউজার কন্ট্রোল। এগুলোর অবস্থান এ্যারো দ্বারা ছবিতে দেখিয়েছি। এদের প্রত্যেকটির বিভিন্ন প্রোপার্টি নিচের টেবিলে তুলে ধরেছি। টুল বাক্স থেকে একে একে এগুলো ফর্মে নিয়ে আসুন। এরপর নির্দেশ মূতাবিক এদের প্রোপার্টি সেট করুন।

নাম	টেবুল প্রোপার্টি	নাম	টেবুল প্রোপার্টি	নাম	টেবুল প্রোপার্টি
BtnBack	<<	BtnAddToFav	বুকমার্ক এই পেইজ	ListBox1	Font: Tahoma, 12pt
BtnForward	>>	BtnClear	ক্লিয়ার	ProgressBar1	--
BtnGo	যান!	TxtURL	--	MyWebBrowser	--
BtnRefresh	ফ্রেশ	TextBox1	ReadOnly: True	Form1	আমার ব্রাউজার
BtnHome	বাড়ি				



১৬তম পরিচ্ছেদ: *Few More Sample Applications*

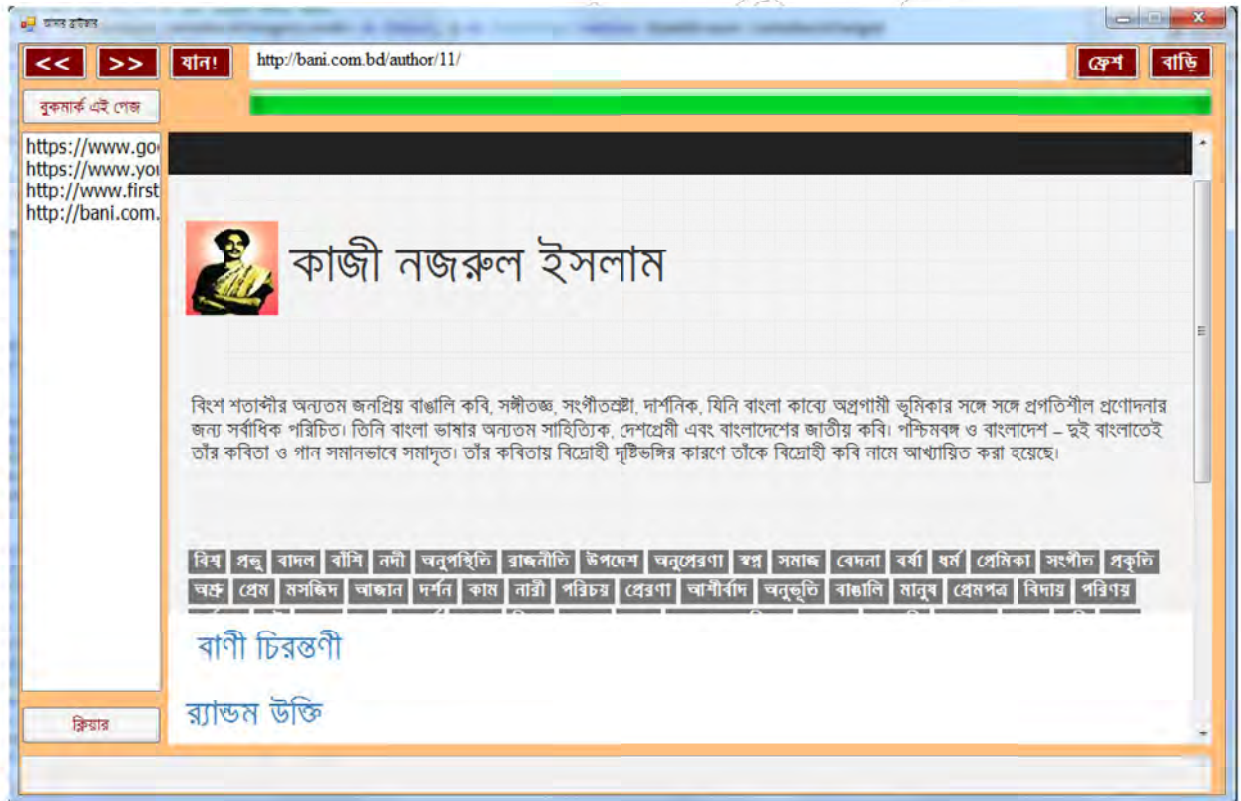
আরো কয়েকটি নমুনা এ্যাপলিকেশন: *Few More Sample Applications*



এসেছি শূন্য লিস্টবাক্সে।

১৩. `Sub_ListBox1_DoubleClick` -এটা হলো লিস্ট বাক্সের ডবলক্লিক ইভেন্ট। একে ধরে আমরা সিলেক্ট করা লাইন মুতাবিক ইউআরএল-এ ন্যাভিগেশন করার জন্য নির্দেশ করেছি ওয়েভব্রাউজারকে।

১৪. `MyWebBrowser_StatusTextChanged` -এটা হলো ওয়েভব্রাউজারের স্টেটাসটেক্সটচেঞ্জড নামক ইভেন্ট সাবস্ক্রিপশন। আমরা একে ধরে প্রদর্শিত ওয়েভ পেইজের ইউআরএল লোড হওয়ার স্টেটাস টেক্সটটুকু নিচের টেক্সট বাক্সে প্রদর্শনের জন্য নিয়ে এসেছি। লোড সম্পন্ন হয়ে গেলে, 'Done' শব্দটি বাক্সে দেখাবে। এবার দেখে নিন আপনার ব্রাউজারের একশন।



১৬তম পরিচ্ছেদ: *Few More Sample Applications*

আরো কয়েকটি নমুনা এ্যাপলিকেশন: *Few More Sample Applications*

'this sub enables aTimer and tTimer and disables wTimer

```
Private Sub wTimer_Tick(sender As Object, e As EventArgs) Handles wTimer.Tick
    wTimer.Enabled = False
    aTimer.Enabled = True
    tTimer.Enabled = True
End Sub
```

এর পরেরটি হলো ফর্ম লোড ইভেন্ট সাবরুটিন। আমরা একে ধরে প্রথমে ডিকুয়ার করা ভেরিয়েবলগুলোতে ভ্যালু প্রবেশ করেছি।

'form load-time initialization of 3 variables associated with three timers

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    aTimer.Interval = speed
    wTimer.Interval = waitTime
    tTimer.Interval = speedText
    ' keep a record of PictureBox Position
    OXpos = MyPictureBox.Location.X : OYPos = MyPictureBox.Location.Y
End Sub
```

নিচের ছোট্ট সাবটি হলো Exit বোতামের ক্লিক ইভেন্ট কোড। বুঝতেই পারছেন, এটি দ্বারা প্রোগ্রাম বন্ধ করা হয়।

'exit- here Me means the Form itself.

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Me.Close()
End Sub
```

ব্যস! কোড মাত্র এটুকুই। এবার টেস্ট রানের পালা। নিচের তিনটি চিত্রে এ্যানিমেশনকালীন স্ক্রিনশট তুলে ধরেছি।



৩. বাংলা-ইংরেজি-বাংলা তারিখ রূপান্তর: **Bangla-English-Bangla Date Conversion**

ইন্টারনেটে একাধিক ওয়েবসাইট আছে যেখানে বাংলা থেকে ইংরেজি এবং ইংরেজি থেকে বাংলা তারিখ রূপান্তরের সুযোগ আছে। কিন্তু কিভাবে এরূপ এ্যাপলিকেশন ডেভেলোপ করা যায়? এ প্রশ্নের জবাব খুঁজতে যেয়েই শেষ পর্যন্ত নিজেই নতুন এই প্রোগ্রাম সৃষ্টি করতে সক্ষম হয়েছি। আর এটি শেয়ার করতে চাই সকল শিক্ষার্থীদের সাথে। এই প্রোগ্রাম ডেভেলোপমেন্ট টেকনিক থেকে ভিজুয়াল বেসিক.নেট প্রোগ্রামিং ভাষা সম্পর্কেও অনেক ব্যাপার জানা ও সেসাথে অভিজ্ঞতা অর্জিত হবে। তাহলে আসুন, দেখে নিই কিভাবে এ ব্যবহারযোগ্য কার্যকরী এ্যাপলিকেশনটি বানাতে হয়।

১৬তম পরিচ্ছেদ: *Few More Sample Applications*

আরো কয়েকটি নমুনা এ্যাপলিকেশন: *Few More Sample Applications*

বিষয়টি স্পষ্ট হবে। মনে করুন এই তারিখ: ২৫/৫/১৮৯৯ [জাতীয় কবি কাজী নজরুল ইসলামের জন্ম তারিখ] বাংলা তারিখে রূপান্তর করবো। আমরা প্রথমেই দেখবো কোন্ সংখ্যাটি বিয়োগ করে বাংলা সন বের করতে হবে। যেহেতু ইংরেজি মাসটি হলো মে [৫] যা ১৫ এপ্রিল থেকে ৩১ ডিসেম্বরের ভেতর পড়ে তাই আমাদেরকে বিয়োগ করতে হবে ৫৯৩ সংখ্যাটি। সুতরাং যে বাংলা সনে কবির জন্ম হয়েছিল তা হলো: ১৮৯৯ - ৫৯৩ = ১৩০৬ বাংলা।

এবার টেবিলের দিকে তাকিয়ে দেখতে পাচ্ছি যে, তারিখের দিনটি ইংরেজি মে (৫ম) মাসের ১৫-৩১ দিনের ভেতর হলে বাংলা তারিখের দিনটি হবে: ইংরেজি তারিখ - ১৪ = ২৫-১৪ = ১১। আর মাস হবে: দ্বিতীয়টি তথা জৈষ্ঠ্য। সুতরাং আমাদের জাতীয় কবির জন্ম হয়েছিল: ১১ জৈষ্ঠ্য, ১৩০৬ বাংলা। সুতরাং যে কোনো ইংরেজি তারিখ বাংলায় রূপান্তরের একটি সূত্র পেয়ে গেলাম। অনুরূপ বাংলা থেকেও ইংরেজিতে রূপান্তর করতে টেবিলের তথ্যাদি যথেষ্ট হবে। যেমন: আজকের [এই লেখা রচনার] তারিখ হলো ২৯ পৌষ, ১৪২৩ বাংলা। আমরা এটাও জানি, আজকের ইংরেজি তারিখ হলো ১২ জানুয়ারী, ২০১৭ ইং। আমরা টেবিল ব্যবহার করে ইংরেজি তারিখ বের করতে পারি এভাবে:

পৌষ মানে বছরের ৯তম মাস। আর তারিখটি হলো ২৯। সুতরাং টেবিলে যেয়ে দেখতে পাচ্ছি যে, ৯ম মাসের ১৮-৩০ দিনের ভেতর হলে বছর পড়ে জানুয়ারী। যেহেতু ১লা জানুয়ারী থেকে ১৩ এপ্রিল পর্যন্ত রেঞ্জের ভেতর তারিখ হলে যোগ দিতে হয় ৫৯৪ তাই ১৪২৩ বাংলা = ১৪২৩ + ৫৯৪ = ২০১৭ ইং। তারিখ বের করতেও আমরা টেবিল ব্যবহার করতে পারি। বাংলা মাস ৯ (পৌষ) ও দিন ১৮-৩০ এর ভেতর হওয়ায় বিয়োগ হবে ১৭। সুতরাং ইংরেজি দিন হবে: ২৯ - ১৭ = ১২। তাই এখন বলতে পারি: ২৯ পৌষ, ১৪২৩ বাংলা = ১২ জানুয়ারী, ২০১৭ ইং।

সুতরাং আমরা যেভাবে উপরের তারিখগুলো পরিবর্তন করেছি তা-ই করতে হবে কম্পিউটারের সফটওয়্যারের মাধ্যমে।

ডিজাইন: ইউজার ইন্টারফেইস ডিজাইন একটি গুরুত্বপূর্ণ কাজ। আপনি চাইবেন ব্যবহারকারী যাতে সহজে এ্যাপলিকেশনটি কাজে লাগাতে পারেন। মোটকথা প্রয়োজনীয় সুযোগ-সুবিধাসহ একটি ব্যবহারকারী-বান্ধব ইউজার ইন্টারফেইস হলো সফল এ্যাপলিকেশন সৃষ্টির চাবিকাঠি। প্রথমে নিচের ডিজাইনটি দেখুন। এরপর কি কি বস্তু টুলবাক্স থেকে আনবেন ও এদের প্রোপার্টি কি হবে তা একটি টেবিলে তুলে ধরবো।

বস্তু	নাম	টেবিল	ফন্ট-সাইজ
ফর্ম	যে কোনো
৪টি বোতাম	যেভাবে আছে	যেভাবে আছে
১৬টি লেবেল	যেভাবে আছে	শেষের ৬টির ভিজিবল=False
৪টি কন্ট্রোল	বাংলা- ১৪
দু'টি টেক্সট বাক্স	বাংলা - ১৪

লক্ষ করুন: উপরের বায়ের কন্ট্রোল নাম: cmbEnglishDay।

ডানেরটির নাম: cmbEnglishMonth। নিচের বায়েরটির নাম:

cmbBanglaDay। ডানেরটি: cmbBanglaMonth। উপরের টেক্সট বাক্সের নাম: txtEnglishYear। নিচেরটির নাম:

txtBanglaYear। এই নামগুলো আমরা কোডে ব্যবহার করেছি। পরিবর্তন বোতামের এ্যারোগুলো আনতে সিঞ্চল কাজে

১৬তম পরিচ্ছেদ: *Few More Sample Applications*

আরো কয়েকটি নমুনা এ্যাপলিকেশন: *Few More Sample Applications*

লাগান। এবার দেখে নিই কোডগুলো।

কোড: আমরা কোডের বিভিন্ন সাবরুটিন ও ফাংশন ব্যাখ্যাসহ আলাদাভাবে তুলে ধরবো। এভাবে হলে কেডগুলো বুঝা অনেকটা সহজ হবে। কারণ, তারিখ পরিবর্তন কোড বাস্তবে তেমন সহজ নয়- বরং বেশ জটিল।

ফর্মের মধ্যে ডবল ক্লিক করে কোড এডিটর খুলুন। ফর্মের ক্লাস ডিক্লারেশন লাইন: **Public Class Form1** -এর নিচে লিখুন এই গ্লোবাল ভেরিয়েবল ডিক্লারেশনগুলো:

Dim ToBanglaAdderVal As Integer 'স্টোর করে ইংরেজি দিনের সাথে কোন্ সংখ্যা যোগ হবে

Dim ToBanglaMonth As Integer 'স্টোর করে বাংলা মাস কি হবে

Dim ToBanglaYear As Integer 'স্টোর করে বাংলা বছর মান কি হবে

Dim IsItLeapYear As Boolean 'বুলিয়ান স্টোরস লিপ ইয়ার কি না - সত্য-মিথ্যা

Dim ToEnglishAdderVal As Integer 'স্টোর করে ইংরেজি দিনের সাথে কোন্ সংখ্যা যোগ হবে

Dim ToEnglishMonth As Integer 'স্টোর করে ইংরেজি মাস কি হবে

Dim ToEnglishYear As Integer 'স্টোর করে ইংরেজি বছর কি হবে

উপরোক্ত ভেরিয়েবলের কোন্টির কাজ কি তা কমেন্ট লাইনে লিখেছি। এবার প্রথম সাবরুটিনটি টাইপ করুন।

```
'প্রোগ্রাম খুললেই যে তারিখ বাংলা ও ইংরেজিতে হবে
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    PutInEnglishDate() 'সাব কল, এটা ইংরেজি আজকের তারিখ বসাবে।
    convertToBanglaDate() 'এ সাব ইংরেজি আজকের তারিখকে বাংলা করে বসাবে।
End Sub
```

আমরা এখানে দু'টি সাবরুটিন কল করেছি যেগুলো পরে ডিফাইন করা হয়েছে। এদের কাজ হলো বাংলা রূপান্তরসহ 'আজকের তারিখ' প্রদর্শন করা। সুতরাং ব্যবহারকারী প্রোগ্রাম রান করার পরই বর্তমান তারিখ ইংরেজি ও বাংলায় দেখতে পাবেন। সাবরুটিন **PutInEnglishDate()** -এর মাধ্যমে উপরের প্রথম কস্বোতে ইংরেজি দিন, পরের কস্বোতে মাস ও টেক্সট বাক্সে বছরের সংখ্যাটি দেখাবে। অনুরূপ নিচের প্রথম কস্বোতে বাংলা দিন, পরের কস্বোতে মাস ও টেক্সট বাক্সে বছরের সংখ্যা প্রদর্শন হবে দ্বিতীয় সাবরুটিন **convertToBanglaDate()** -এর মাধ্যমে। এগুলো কিভাবে কাজ করে তার ওপর আলোচনা সময়মতো হবে। পরের সাবটি হলো ইংরেজি থেকে বাংলায় পরিবর্তন বোতামের ক্লিক ইভেন্ট কোড।

```
'এটা ইংরেজি থেকে বাংলায় পরিবর্তনের বোতাম।
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    If Val(txtEnglishYear.Text) < 100 Or Val(txtEnglishYear.Text) > 9999 Then
        MsgBox("বছর রেঞ্জ হবে ১০০ থেকে ৯৯৯৯।")
        txtEnglishYear.Text = ""
        txtEnglishYear.Focus()
    Exit Sub
End If
Label15.Text = ""
'লিপ ইয়ার না হলে ফেব্রুয়ারি ২৯ এর নিচে থাকতে হবে।
If IsItLeap(Val(txtEnglishYear.Text)) = False Then
    If (cmbEnglishDay.SelectedItem) >= 29 And (cmbEnglishMonth.SelectedItem) = 2 Then
        MsgBox("Not a leap year! February month has 28 days only.", MsgBoxStyle.Exclamation)
    Exit Sub
End If
Else
    'যদি লিপ ইয়ার হয়ে থাকে তাহলে ফেব্রুয়ারি অবশ্যই ২৯ কিংবা কম থাকতে হবে।
    If (cmbEnglishDay.SelectedItem) >= 30 And (cmbEnglishMonth.SelectedItem) = 2 Then
        MsgBox("Its a leap year! February month has 28 days only.", MsgBoxStyle.Exclamation)
    Exit Sub
```

১৬তম পরিচ্ছেদ: *Few More Sample Applications*

আরো কয়েকটি নমুনা এ্যাপলিকেশন: *Few More Sample Applications*

```

End If
End If
'লিপ ইয়ার নয় বা লিপ ইয়ার হয়, এর পরও দিনের তারিখ সঠিক থাকলে
UpdateLabel() 'গোপন লেবেলে দিন, মাস ও বছরের মান নিয়ে যেতে হবে, এই সাবের কাজ এ টি।
'এরপর দেখতে হবে দিনের মাস অন্যান্য মাসের বেলায়ও সঠিক আছে কি না।
If IsDayEntryCorrect((Val(Label10.Text)), Val(Label9.Text)) = False Then
    MsgBox("উপস! মাসে য় দিন আছে তারচেয়ে বেশি হয়ে গেছে, বোধ হয়!", MsgBoxStyle.Exclamation)
Exit Sub
End If
'এবার সব সঠিক হয়েছে
convertToBanglaDate() 'কল কনভার্ট টু বাংলা সাবরুটিন।
End Sub

```

কোডের ব্যাখ্যা: প্রথম ইফ-এন্ড ইফ ব্লকের কাজ হলো বছরের সংখ্যাটি রেঞ্জের ভেতর আছে কি না তা নিশ্চিত করা। আমরা ইংরেজি ১০০ থেকে ৯৯৯৯ সাল পর্যন্ত রেঞ্জ দিয়েছি। এর বাইরে হলে প্রোগ্রাম মেসেজ দিয়ে কোড থেকে বের হবে। এরপর আমরা লেবেল ১৫ এর টেক্সট মুছে দিয়েছি। পরের ইফ-এন্ড ইফ ব্লকে আরেকটি টেস্ট করা হয়েছে। আমরা জানতে চাই, বছর সংখ্যাটি লিপ ইয়ার কি না। এ চেকটি করেছি একটি ফাংশন দ্বারা, যার নাম IsItLeap। এর একটিমাত্র প্যারামিটার আছে যা হলো বছর সংখ্যাটি। ফাংশন বুলিয়ান [হ্যাঁ, না] রিটার্ন করে। সুতরাং দেখতে হবে রিটার্ন করার পরও ব্যবহারকারী ফেব্রুয়ারি মাসের জন্য ২৮ দিন বা তার নিচে প্রবেশ করেছেন কি না। পরের ইফ-এন্ড ইফ ব্লকের কাজ এটি। কিন্তু লিপ ইয়ার হয়ে থাকলে দিন সংখ্যা যাতে ২৯ এর বেশি না হয় তা-ও দেখতে হবে। সুতরাং একই ইফ-এন্ড ইফ ব্লকেই Else কিওয়ার্ড দ্বারা আরো একটি চেক করতে হয়েছে। উভয় চেক সফল হওয়ার পরই পরবর্তী কোড এক্সেকিউট হতে পারে। প্রথমে আমরা একটি সাবরুটিন [UpdateLabel()] কল করে তারিখের সংখ্যাগুলো লুকানো তিনটি লেবেলে নিয়ে গেছি। এরপর আরো একটি সংখ্যা পরীক্ষা করা দরকার। মাসের মধ্যে য'দিন আছে তারচে বেশি প্রবেশ হয়েছে কি না তা-ও জানতে হবে। এই টেস্ট সফল হওয়ার পর আমরা অবশেষে প্রস্তুত হলাম পরিবর্তন সাবরুটিন কল করতে। এই সাবরুটিনের নাম convertToBanglaDate()। সঠিক সময়ে এটার ব্যাখ্যা আসবে।

ইতোমধ্যে বলেছি, আগের সাবরুটিনে UpdateLabel() নামক একটি সাবরুটিন কল করেছি। সুতরাং এটাই আমাদের পরবর্তী সাবরুটিন। এর কাজ হলো কনসোল থেকে ইংরেজি দিনের সংখ্যা, ইংরেজি মাসের সংখ্যা ও [টেক্সট বাক্স থেকে] ইংরেজি বছরের সংখ্যা যথাক্রমে লেবেল ১৯, ১০ ও ১১ -তে কপি করে রাখা।

এর পরের ফাংশনটির কথাও ইতোমধ্যে বলেছি। এটা হলো IsItLeap নামক বুলিয়ান ফাংশন। এর কাজ হচ্ছে বছরটি লিপ ইয়ার কি না পরীক্ষা করে সত্য-মিথ্যা ফলাফল রিটার্ন করা। লক্ষ করুন, কোনো সংখ্যার ভাগফলে অবশিষ্ট আছে কি না জানতে হলে সিস্টেমের মড [Mod] ফাংশন ব্যবহার করতে হয়। সুতরাং এবার টাইপ করুন এ দু'টো প্রসিডোর।

```

Private Sub UpdateLabel() 'লেবেল আপডেট কোড।
    Label9.Text = cmbEnglishDay.SelectedItem 'কনসোল থেকে লেবেলে
    Label10.Text = cmbEnglishMonth.SelectedItem 'কনসোল থেকে লেবেলে
    Label11.Text = txtEnglishYear.Text 'টেস্ট বাক্স থেকে লেবেলে
End Sub
'check if it is a Leap year
Private Function IsItLeap(TheYear As Integer) As Boolean
    If TheYear Mod 4 = 0 Then 'মড রিটার্ন করে ভাগের অবশিষ্ট। না থাকলে ০
        Return True 'এটা লিপ ইয়ার
    Else
        Return False 'এটা লিপ ইয়ার নয়
End If

```


১৬তম পরিচ্ছেদ: *Few More Sample Applications*

আরো কয়েকটি নমুনা এ্যাপলিকেশন: *Few More Sample Applications*

```

Return "-15" & " 4"
End If
If MonthInt = 8 And DayInt >= 1 And DayInt <= 15 Then
Return "16" & " 4"
End If
If MonthInt = 8 And DayInt >= 16 And DayInt <= 31 Then
Return "-15" & " 5"
End If
If MonthInt = 9 And DayInt >= 1 And DayInt <= 15 Then
Return "16" & " 5"
End If
If MonthInt = 9 And DayInt >= 16 And DayInt <= 30 Then
Return "-15" & " 6"
End If
If MonthInt = 10 And DayInt >= 1 And DayInt <= 15 Then
Return "15" & " 6"
End If
If MonthInt = 10 And DayInt >= 16 And DayInt <= 31 Then
Return "-15" & " 7"
End If
If MonthInt = 11 And DayInt >= 1 And DayInt <= 14 Then
Return "16" & " 7"
End If
If MonthInt = 11 And DayInt >= 15 And DayInt <= 30 Then
Return "-14" & " 8"
End If
If MonthInt = 12 And DayInt >= 1 And DayInt <= 14 Then
Return "16" & " 8"
End If
If MonthInt = 12 And DayInt >= 15 And DayInt <= 31 Then
Return "-14" & " 9"
End If
Return 0
End Function

```

ইতোমধ্যে যে ফাংশনটির ওপর বিস্তারিত বলা হয়েছে তা থেকে আমরা বাংলা দিন ও মাস পাবো মাত্র। তবে আমাদেরকে তো বাংলা বছর কি হবে তা-ও জানতে হবে। এটা জানার জন্য পরের এই ফাংশনটি।

```

Private Function FindBanglaYear(EngYear As Integer,
EngMonth As Integer, DateDay As Integer) As Integer
If EngMonth >= 1 And EngMonth <= 3 Then
ToBanglaYear = EngYear - 594 'because
banglaYear 1 less, no Boishak yet
Return ToBanglaYear
End If
If EngMonth = 4 And DateDay >= 1 And
DateDay <= 13 Then
ToBanglaYear = EngYear - 594 'because
banglaYear 1 less, no Boishak yet
Return ToBanglaYear
End If
ToBanglaYear = EngYear - 593 'Here it change
Return ToBanglaYear
End Function

```

লক্ষ করুন, উপরের ফাংশনে তিনটি প্যারামিটার আছে। এগুলো হলো যথাক্রমে ইংরেজি বছর, মাস ও দিন। প্রতিটি ভ্যালু হবে ইন্টেজার। ফাংশন রিটার্ন করবে বাংলা বছর [ইন্টেজার হিসাবে]। আমরা জানি ১লা বৈশাখ না আসা পর্যন্ত বাংলা বছর সংখ্যা বাড়বে না। এ কারণেই প্রথমে ইংরেজি মাস ১ থেকে ৩ এর ভেতর হলে বাংলা মাস বিয়োগ করেছি ৫৯৪ সংখ্যা দ্বারা। বাকি মাসগুলোর ক্ষেত্রে বিয়োগ হবে ৫৯৩।

আমাদের ফর্মে উপরের বায়ের বোতামটি হলো বর্তমান তারিখ বাংলা ও ইংরেজি তারিখ প্রদানের বোতাম। সুতরাং এটার ক্লিক ইভেন্ট হবে নিম্নরূপ।

```

'English to Bangla Date conversion button sub
Private Sub Button2_Click(sender As Object, e As
EventArgs) Handles Button2.Click
Label15.Text = ""
PutInEnglishDate() 'পরিবর্তন [ইংরেজি থেকে বাংলা]
বোতাম। কল সাব।
convertToBanglaDate() 'কল সাব পরিবর্তন করতে।
End Sub

```


১৬তম পরিচ্ছেদ: *Few More Sample Applications*

আরো কয়েকটি নমুনা এ্যাপলিকেশন: *Few More Sample Applications*

'update Bangla label

Private Sub UpdateBLabel() 'লেবেল আপডেট কোড।

Label12.Text = cmbBanglaDay.SelectedItem 'কম্বো থেকে লেবেলে

Label13.Text = cmbBanglaMonth.SelectedItem 'কম্বো থেকে লেবেলে

Label14.Text = txtBanglaYear.Text 'টেক্সট বাক্স থেকে লেবেলে

End Sub

আমরা ইতোমধ্যে বলেছি, বাংলা তারিখ ঠিক আছে কি না পরীক্ষার জন্য একটি ফাংশন আছে। নিচের কোডগুলোই হচ্ছে এ ফাংশন। এটি লিপ ইয়ার ও সঠিক তারিখ, মাসের সঙ্গে তারিখের সামঞ্জস্য ইত্যাদি পরীক্ষা করে সত্য বা মিথ্যা ফলাফল হিসাবে রিটার্ন করে।

'check to see if the dateEntry correct

Private Function IsBanglaDateCorrect(BDay As Integer, BMonth As Integer, BYear As Integer) As Boolean

Dim CheckLeap As Boolean = IsItLeap(BYear)

If CheckLeap = True Then

If BMonth = 11 And BDay <= 31 Then

Return True

End If

If BMonth >= 1 And BMonth <= 5 Then

If BDay <= 31 Then

Return True

End If

End If

If BMonth >= 6 And BMonth <= 10 Then

If BDay <= 30 Then

Return True

End If

End If

If BMonth = 12 And BDay <= 30 Then

Return True

End If

End If

'leap year false

If BMonth >= 1 And BMonth <= 5 Then

If BDay <= 31 Then

Return True

End If

End If

If BMonth >= 6 And BMonth <= 11 Then

If BDay <= 30 Then

Return True

End If

End If

If BMonth = 12 And BDay <= 30 Then

Return True

End If

Return False

End Function

পরের সাবরুটিনের কাজ হলো বাংলা তারিখকে ইংরেজি তারিখে রূপান্তর করা।

১৬তম পরিচ্ছেদ: *Few More Sample Applications*

আরো কয়েকটি নমুনা এ্যাপলিকেশন: *Few More Sample Applications*

'bangla to english date conversion

Private Sub convertToEnglishDate()

Dim BanglaDay As Integer = Val(Label12.Text) 'গোপন লেবেল থেকে সংখ্যা নিয়ে আসা

Dim BanglaMonth As Integer = Val(Label13.Text) 'গোপন লেবেল থেকে সংখ্যা নিয়ে আসা

Dim BanglaYear As Integer = Val(Label14.Text) 'গোপন লেবেল থেকে সংখ্যা নিয়ে আসা

Dim DayAdderMonth As String = FindEnglishRange(BanglaMonth, BanglaDay) 'ফাংশন কল

Dim Words As String() = DayAdderMonth.Split(New Char() {" "})

ToEnglishAdderVal = Words(0)

ToEnglishMonth = Words(1)

ToEnglishYear = FindEnglishYear(BanglaYear, BanglaMonth, BanglaDay) 'ফাংশন কল

Label9.Text = BanglaDay + (ToEnglishAdderVal) 'এড সঠিক মান বাংলা দিন পেতে।

Label10.Text = ToEnglishMonth 'গোপন লেবেলে মাস মান সংরক্ষণ

cmbEnglishDay.SelectedItem = Label9.Text 'গোপন লেবেল থেকে কন্সোলতে নেওয়া

cmbEnglishMonth.SelectedItem = Label10.Text 'গোপন লেবেল থেকে কন্সোলতে নেওয়া

txtEnglishYear.Text = ToEnglishYear 'বাংলা বছরের ফিল্ডে মান নেওয়া

Label16.Text = cmbEnglishMonth.SelectedItem & "/" & Label9.Text & "/" & txtEnglishYear.Text

Label16.Text = FormatDateTime(Label16.Text, DateFormat.LongDate)

Label15.Text = getBanglaDayMothName(Label16.Text, Val(Label12.Text), Val(Label13.Text), Val(Label14.Text)) 'দিনের নামসহ তারিখ

End Sub

এই সাবরুটিন দু'টি গুরুত্বপূর্ণ ফাংশন কল করে। এগুলো সম্পর্কে ব্যাখ্যা একটু পরই আসছে। এখানে যা বলার তাহলো, প্রথম ফাংশন রিটার্ন করে একটি স্ট্রিংয়ের মধ্যে দু'টি মান। প্রথমটি হলো ইংলিশ দিনের সাথে কতো যোগ বা বিয়োগ করতে হবে। আর দ্বিতীয়টি হলো বাংলা মাসটি কি হবে। আমরা স্ট্রিংকে সিস্টেমের Split নামক ফাংশন দ্বারা আলাদা দু'টি শব্দে বিভক্ত করে মানগুলো বের করে যথাক্রমে ToEnglishAdderVal এবং ToEnglishMonth ভেরিয়েবলে সংরক্ষণ করেছি। দ্বিতীয় ফাংশন থেকে প্রাপ্ত ইংলিশ বছরের মান সরাসরি টেক্সটবক্সে নিয়ে গেছি প্রদর্শনের জন্য। লক্ষ করুন টেক্সট বক্স ১৫ ও ১৬-তে দিনের ও মাসের নামসহ যথাক্রমে ইংরেজি ও বাংলা তারিখটি বসিয়েছি। ঠিক যেমনটি করেছিলাম ইংরেজি থেকে বাংলায় তারিখ পরিবর্তনের সময়। এবার আরো একটি ফাংশনের কোড দেখবো।

'returns bangla year

Private Function FindEnglishYear(BangYear As Integer, BangMonth As Integer, DateDay As Integer) As Integer

If BangMonth >= 1 And BangMonth <= 8 Then

ToEnglishYear = BangYear + 593 'because banglaYear 1 less, no Boishak yet

Return ToEnglishYear

End If

If BangMonth = 9 And DateDay >= 1 And DateDay <= 17 Then

ToEnglishYear = BangYear + 593 'because banglaYear 1 less, no Boishak yet

Return ToEnglishYear

End If

ToEnglishYear = BangYear + 594 'Here it change

Return ToEnglishYear

End Function

আমাদেরকে ইংরেজি বছরের মান বের করতে হবে। উপরের ফাংশনের কাজ এটি। লক্ষ করুন, বাংলা মাস যদি ১ থেকে ৮ এর ভেতর হয় তাহলে যোগ করতে হবে ৫৯৩। একইভাবে বাংলা মাস ৯ [ফাল্গুন] হলে ও দিনের মান ১ থেকে ১৭ পর্যন্ত হলে যোগ হবে ৫৯৩। অন্যথায় যোগ হবে ৫৯৪।

১৬তম পরিচ্ছেদ: *Few More Sample Applications*

আরো কয়েকটি নমুনা এ্যাপলিকেশন: *Few More Sample Applications*

এবার যে ফাংশনটি দেখাবো তা-ই প্রোগ্রামের সর্বশেষ কোড ব্লক। অবশ্য ডানের উপরের বোতামের কোডও আরেকটি আছে। এটা প্রোগ্রামার সম্পর্কে জ্ঞাতব্য ব্যাপার মাত্র।

'returns what number to be added [or subtracted] to get correct english day value

Private Function FindEnglishRange(MonthInt As Integer, DayInt As Integer) As String

If MonthInt = 1 And DayInt >= 1 And DayInt <= 17 Then
Return "13" & " 4"

End If

If MonthInt = 1 And DayInt >= 18 And DayInt <= 31 Then
Return "-17" & " 5"

End If

If MonthInt = 2 And DayInt >= 1 And DayInt <= 17 Then
Return "14" & " 5"

End If

If MonthInt = 2 And DayInt >= 18 And DayInt <= 31 Then
Return "-17" & " 6"

End If

If MonthInt = 3 And DayInt >= 1 And DayInt <= 16 Then
Return "14" & " 6"

End If

If MonthInt = 3 And DayInt >= 17 And DayInt <= 31 Then
Return "-16" & " 7"

End If

If MonthInt = 4 And DayInt >= 1 And DayInt <= 16 Then
Return "15" & " 7"

End If

If MonthInt = 4 And DayInt >= 17 And DayInt <= 31 Then
Return "-16" & " 8"

End If

If MonthInt = 5 And DayInt >= 16 And DayInt <= 15 Then
Return "15" & " 8"

End If

If MonthInt = 5 And DayInt >= 17 And DayInt <= 31 Then
Return "-16" & " 9"

End If

If MonthInt = 6 And DayInt >= 1 And DayInt <= 15 Then
Return "15" & " 9"

End If

If MonthInt = 6 And DayInt >= 16 And DayInt <= 30 Then
Return "-15" & " 10"

End If

If MonthInt = 7 And DayInt >= 1 And DayInt <= 16 Then
Return "15" & " 10"

End If

If MonthInt = 7 And DayInt >= 17 And DayInt <= 30 Then
Return "-16" & " 11"

End If

If MonthInt = 8 And DayInt >= 1 And DayInt <= 16 Then
Return "14" & " 11"

End If

If MonthInt = 8 And DayInt >= 17 And DayInt <= 30 Then
Return "-16" & " 12"

End If

If MonthInt = 9 And DayInt >= 1 And DayInt <= 17 Then
Return "14" & " 12"

End If

If MonthInt = 9 And DayInt >= 18 And DayInt <= 30 Then
Return "-17" & " 1"

End If

If MonthInt = 10 And DayInt >= 1 And DayInt <= 18 Then
Return "13" & " 1"

End If

If MonthInt = 10 And DayInt >= 19 And DayInt <= 30 Then
Return "-18" & " 2"

End If

If MonthInt = 11 And DayInt >= 1 And DayInt <= 17 Then
Return "12" & " 2"

End If

If MonthInt = 11 And DayInt >= 17 And DayInt <= 31 Then
Return "-16" & " 3"

End If

If MonthInt = 12 And DayInt >= 1 And DayInt <= 17 Then
Return "14" & " 3"

End If

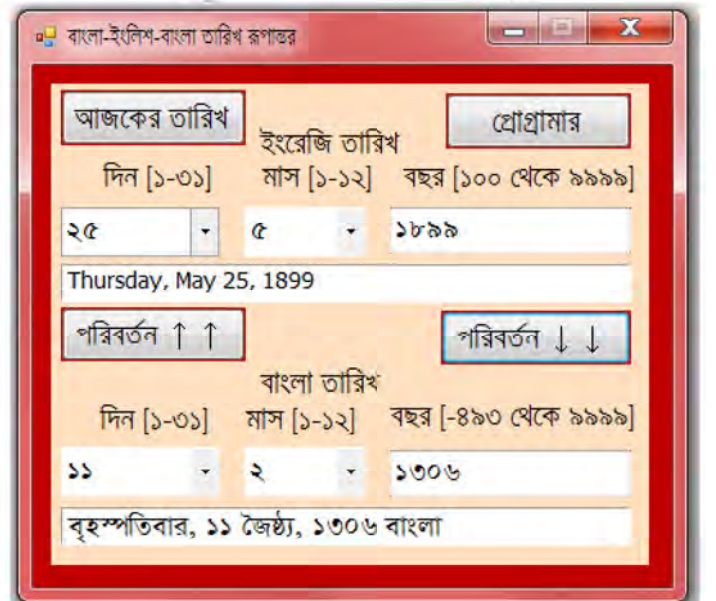
If MonthInt = 12 And DayInt >= 18 And DayInt <= 30 Then
Return "-17" & " 4"

End If

Return 0

End Function

এখানেই ফাংশনের শেষ। এটি কাজ করে আগে বর্ণিত FindDayRange নামক ফাংশনের মতো। ফাংশনের দুটি প্যারামিটার আছে। তাহলো, ইংরেজি মাসের মান ও দিনের মান। কোন্ মাস ও কোন দিন তা-ই নির্দিষ্ট করে বাংলা কোন্ মাস ও দিন হবে। আগেরটির মতো এই ফাংশনও একটি স্ট্রিং রিটার্ন করে যেখানে দুটি সংখ্যা থাকবে। আমরা কলিং সারকটিনে Split নামক সিস্টেম ফাংশন দ্বারা আলাদা করেছি। এরপর উভয় সংখ্যা কাজে লাগিয়েছি। যাক, কোডের ব্যাখ্যা এখানেই শেষ। এবার রান করে দেখুন [নিচের চিত্র]।



১৭তম পরিচ্ছেদ: *Windows Form Controls*

উইন্ডোজ ফর্ম কন্ট্রোলস: *Windows Form Controls*

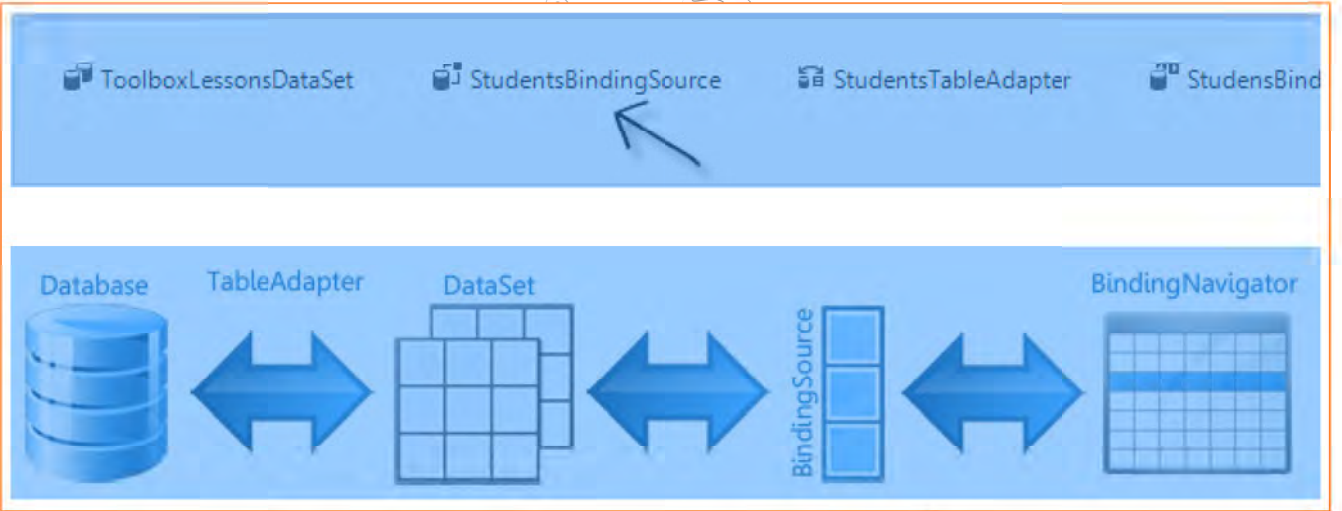
1. Pointer: টুলস বাক্সের প্রথম সদস্য হলেও এটা বাস্তবে কোনো কন্ট্রোল নয়! তবে ফর্মে কন্ট্রোল নেওয়ার জন্য এটি কাজ করে। এছাড়া সিলেক্ট করা কোনো কন্ট্রোলকে ডিসিলেক্ট করতেও এটি ব্যবহৃত হয়।

2. BackgroundWorker: এটা বেকগ্রাউন্ড বা নেপথ্যে কাজ করে- কিন্তু ফর্মের ওপর এর কোনো এফেক্ট নেই। অথচ ফর্মের কোড থেকেই একে নির্দেশনা দেওয়া যায়। দীর্ঘ কোনো অপারেশনে বেকগ্রাউন্ডওয়ারকার খুব ব্যবহারোপযোগী।

3. BindingNavigator: এটা ডাটা রেকর্ডের মধ্যে ন্যাভিগেট করার জন্য ব্যবহৃত হয়। আমরা যখন ভিজুয়াল স্টুডিও এর উইজার্ড দ্বারা ডাটাবেইজ প্রোগ্রাম সৃষ্টি করেছিলাম, তখন এমনিতেই উপরে কয়েকটি ন্যাভিগেশন বোতাম এসেছিল। এগুলো এই বাইন্ডিংন্যাভিগেটরের কাজ। এটি ফর্মের নিচে ধূসর এলাকায় পড়ে থাকে। চিত্রটি দেখুন।

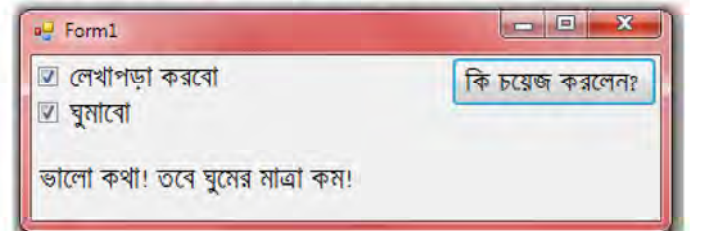
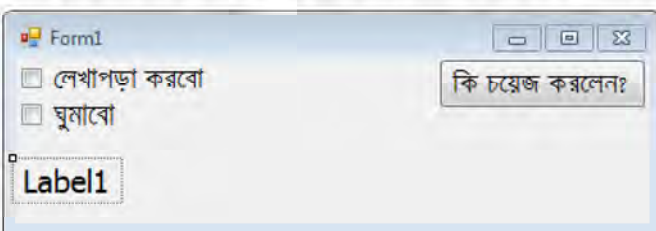


4. BindingSource: এটার কাজ হলো ফর্মের মধ্যে সম্পর্কিত ডাটা নিয়ন্ত্রণ করা। ভিজুয়াল বেসিক.নেট ডাটাবেইজের ডাটা প্রদর্শন ও নিয়ন্ত্রণে সাধারণত চারটি অবজেক্ট বা কন্ট্রোল ব্যবহার করে: 1. DataSet, 2. TableAdapter, 3. BindingSource & 4. BindingNavigator। দেখুন নিচের ডায়াগ্রামটি।



5. Button: এটা সম্পর্কে বেশি বলার প্রয়োজন নেই- শুধু এটুকু ছাড়া: প্রায় সকল এ্যাপলিকেশনেই এই অবজেক্টটি ব্যবহৃত হয়। এ গ্রন্থের প্রতিটি উদাহরণে এক বা ততোধিক বোতাম ব্যবহৃত হয়েছে।

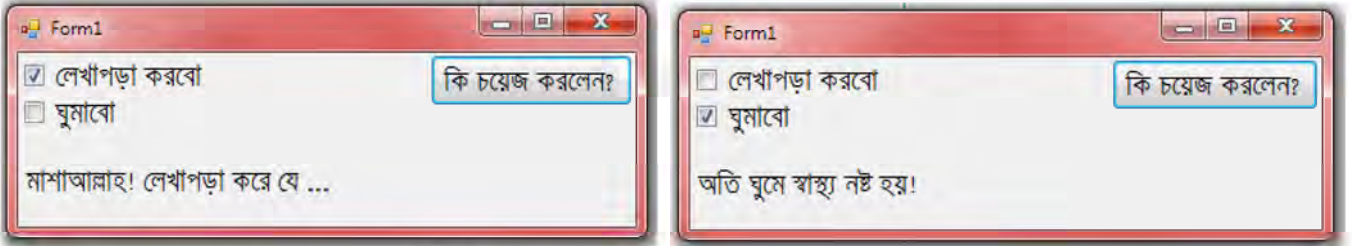
6. CheckBox: এ বস্তুটি আমরা খুব বেশি ব্যবহার করি নি। এটা মূলত একটি বাক্স যেখানে ব্যবহারকারী ক্লিক করলে একটি চেকমার্ক আসে। পুনরায় ক্লিক করলে তা চলে যায়। বাক্সের ডানে [আপনার নিজস্ব] যা লেখা থাকবে তা অত্যন্ত গুরুত্বপূর্ণ। যেমন আপনি লিখতে পারেন: Volume এবং আরেকটিতে লিখতে পারেন: Area। এখন ব্যবহারকারী উভয়টি কিংবা যে কোনো একটি কিংবা কোনোটি নয়- এভাবে চেকমার্ক করতে পারেন। এ থেকে আপনি জানলেন কোনটি চয়েজ করা হয়েছে ও প্রয়োজনীয় একশন নিলেন কোডের মাধ্যমে। একটি ছোট্ট উদাহরণ দেখুন। বায়ের চিত্রে



১৭তম পরিচ্ছেদ: *Windows Form Controls*

উইন্ডোজ ফর্ম কন্ট্রোলস: *Windows Form Controls*

ডিজাইন তুলে ধরেছি ডানেরটিতে উভয় বোতাম চয়েজ করার পর কী মেসেজ লেবেলে আসবে তা দেখা যাচ্ছে। বোতামের কোডে আরো দু'টি চয়েজের জন্য দু'টি ভিন্ন মেসেজও আছে। নিচের চিত্রদ্বয় দেখুন।



ফর্মের 'কি চয়েজ করলেন?' বোতামের ক্লিক ইভেন্ট সাবে লিখুন:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    If CheckBox1.Checked = True Then
        Label1.Text = "মাশাআল্লাহ! লেখাপড়া করে যে ..."
    End If
    If CheckBox2.Checked = True Then
        Label1.Text = "অতি ঘুমে স্বাস্থ্য নষ্ট হয়!"
    End If
    If CheckBox1.Checked = True And CheckBox2.Checked = True Then
        Label1.Text = "ভালো কথা! তবে ঘুমের মাত্রা কম!"
    End If
    If CheckBox1.Checked = False And CheckBox2.Checked = False Then
        Label1.Text = "আরে! ঘুমাবেনও না, পড়বেনও না! মনে হয় আপনি লেজি!!"
    End If
End Sub
```

7. CheckedListBox: এটা উপরে CheckBox ব্যাখ্যাকৃত কন্ট্রোলের মতোই কাজ করে। তবে অবশ্যই কয়েকটি মৌলিক পার্থক্য আছে। একটি উদাহরণ দেখি।

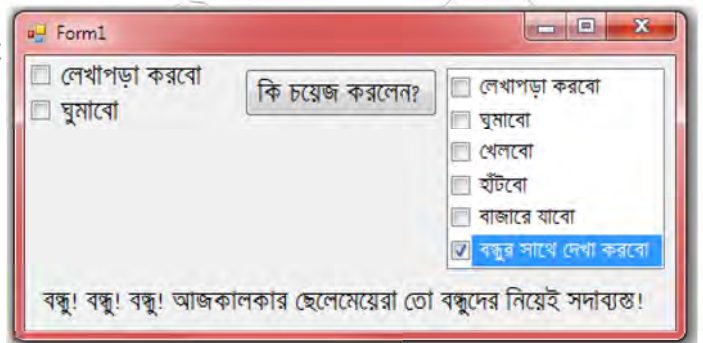
ফর্মের মধ্যে একটি CheckedListBox control যোগ করুন। কন্ট্রোল উপর রাইট ক্লিক করে চয়েজ করুন: Edit Items ...। একটি জানালা আসবে এতে লিস্ট লিখুন:

```
☑ ( ) ☐ ☐ ☐ ☐ ☐
*☐ ☐ ☐ ☐ ☐
☑ ( ☐ ☐ ☐
☐ ☐ ☐ ☐ ☐ ☐ ☐
☐, ☐ ☐ ☐ ☐ ( ☐ ☐ ☐ ☐
☐
```

Ok বোতামে ক্লিক করুন। দেখবেন লিস্টটি এসে গেছে।

এবার নিচের কোড ব্লকটি চেকডলিস্টবাক্সের মধ্যে ক্লিক করার পর যে সাবরুটিন আসবে তার ভেতর লিখুন। রান করুন, টেস্ট করুন।

```
Private Sub CheckedListBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
CheckedListBox1.SelectedIndexChanged
    With CheckedListBox1
        If .SelectedIndex = 0 Then
            Label1.Text = "... গাড়ি ষোড়া চড়বে! জ্ঞানী হবে।"
        End If
        If .SelectedIndex = 1 Then
```



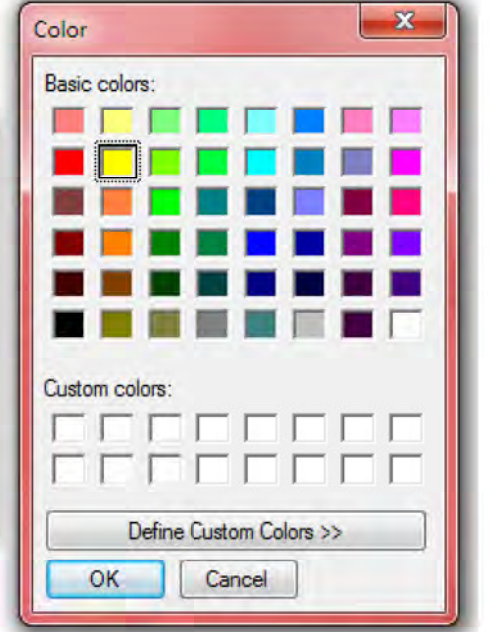
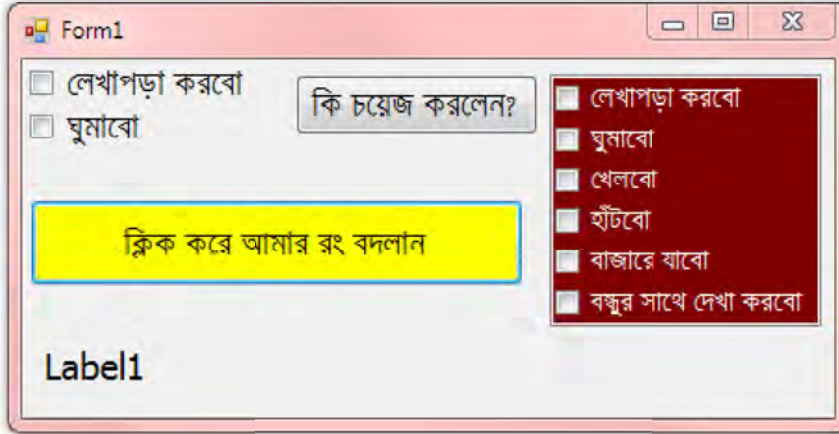
১৭তম পরিচ্ছেদ: *Windows Form Controls*

উইন্ডোজ ফর্ম কন্ট্রোলস: *Windows Form Controls*

```

Label1.Text = "ঠিক আছে! সাবইকেই ঘুমোতে হয়- তবে অতিরিক্ত নয়!"
End If
If .SelectedIndex = 2 Then
    Label1.Text = "খেলাধুলো শরীরের জন্য ভালো বটে! কিন্তু বেশি না!"
End If
If .SelectedIndex = 3 Then
    Label1.Text = "অকে! হাঁটলে ব্যায়াম হয়। ব্যায় হলে স্বাস্থ্য ভালো থাকে!"
End If
If .SelectedIndex = 4 Then
    Label1.Text = "খুব তাড়াতাড়ি ফিরে এসো! বধুয়া অপেক্ষা করিতেছে, রাঁধিবে!"
End If
If .SelectedIndex = 5 Then
    Label1.Text = "বন্ধু! বন্ধু! বন্ধু! আজকালকার ছেলেমেয়েরা তো বন্ধুদের নিয়েই সদাব্যস্ত!"
End If
End With
End Sub
    
```

8. ColorDialog: এটা হচ্ছে রং করার ডায়ালগ প্রদর্শনের কন্ট্রোল। নিচের উদাহরণটি দেখুন। উপরের ফর্মে এরকটি বোতাম ও একটি কালার ডায়ালগ কন্ট্রোল টুলবাক্স থেকে যোগ করুন। বোতামের ক্লিক ইভেন্ট সাবরুটিনে লিখুন:



```

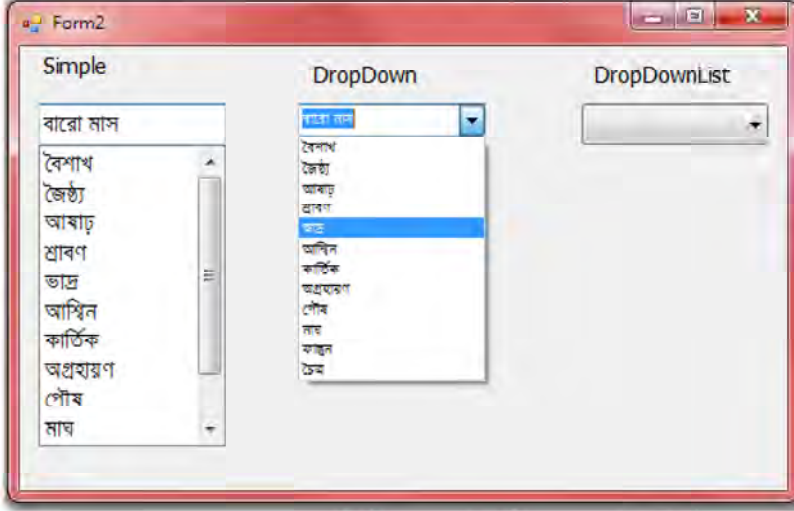
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    If ColorDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
        Button2.BackColor = ColorDialog1.Color
    End If
End Sub
    
```

উপরের চিত্রে রানকালীন অবস্থা তুলে ধরেছি। ডানের ডায়ালগটিই হলো **ColorDialog**। আপনি যে কোনো রঙে ক্লিক করে OK দিন।

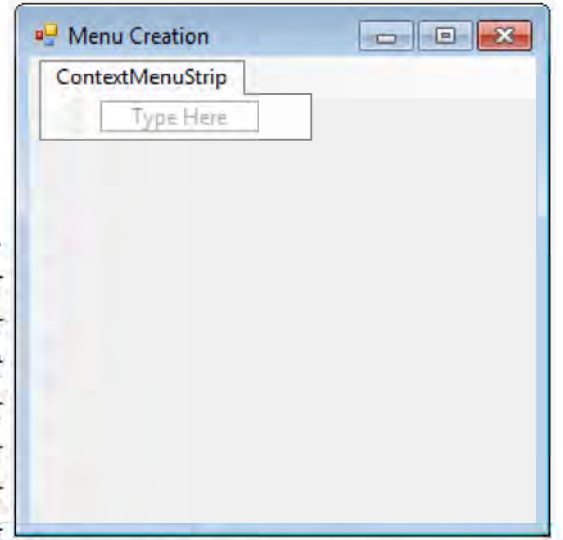
১৭তম পরিচ্ছেদ: *Windows Form Controls*

উইন্ডোজ ফর্ম কন্ট্রোলস: *Windows Form Controls*

9. ComboBox: এটির ব্যবহার আমরা দেখেছি। তিন ধরনের কন্ট্রোল আছে: Simple, DropDown and DropDownList।

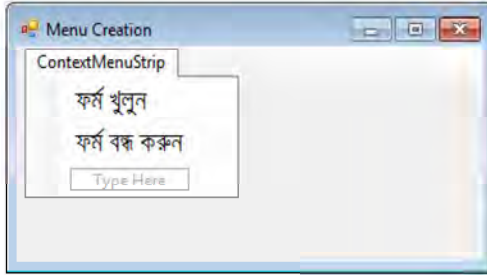


বায়ের চিত্রে তিন ধরনের কন্ট্রোল দেখা যাচ্ছে। লক্ষণীয় যে, টুলবাক্সে কন্ট্রোল একটাই। কিন্তু প্রোপার্টিতে এর ধরন নির্ধারণ করা যায়। এই প্রোপার্টির নাম `DropDownStyle`। এখানেই তিনটির যে কোনো একটি চয়েজ করতে পারেন।



10. ContextMenuStrip: এটা মাউসে রাইট-ক্লিক [Right-Click] করে মেনু তৈরির অবজেক্ট। ফর্মে নিলেই এটা মেনু তৈরির

[ডিজাইনের] জন্য প্রস্তুতি নেবে। পাশের চিত্রটি দেখুন। এরপর টাইপ করে আপনার মেনু আইটেম যোগ করতে পারেন। লক্ষ করুন, আমাদের উদাহরণে



বাংলায় মেনু আইটেম লিখেছি। ইউনিকোডে লিখলে কোনো সমস্যা নেই। এ

উভয় মেনু আইটেমের ক্লিক ইভেন্ট হ্যান্ডলারে কোড লিখে আমরা নির্দেশ দিতে পারি। আমাদের উদ্দেশ্য ব্যবহারকারী প্রথম মেনু আইটেম চয়েজ করলে একটি ফর্ম খুলবে। দ্বিতীয়টি দ্বারা তা বন্ধ হয়ে যাবে। প্রথমে আমাদেরকে অবশ্য সেই ফর্মটি প্রজেক্টে যুক্ত করতে হবে। সুতরাং প্রজেক্ট মেনু থেকে একটি নতুন ফর্ম যুক্ত করুন। এবার এই কোডগুলো প্রথম ফর্মের ক্লাসের ভেতর লিখুন। মনে রাখবেন ফর্মের সাথে এই মেনুকে জড়িত করতে হবে। এটা করতে ফর্মে ক্লিক করুন। প্রোপার্টিতে যেয়ে `ContextMenuStrip` এর ডানে (none) শব্দের উপর ক্লিক করলেই দেখতে পাবেন `ContextMenuStrip1`। এটি চয়েজ করুন।

`Private Sub ফর্মখুলুন_Click(sender As Object, e As EventArgs) Handles ফর্মখুলুন.Click`

```

If Form2.Visible = False Then
    Form2.Show()
End If
End Sub

```

`Private Sub ফর্মবন্ধকরুন_Click(sender As Object, e As EventArgs) Handles ফর্মবন্ধকরুন.Click`

```

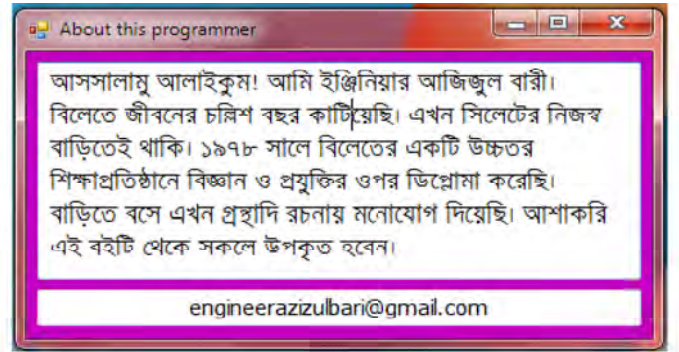
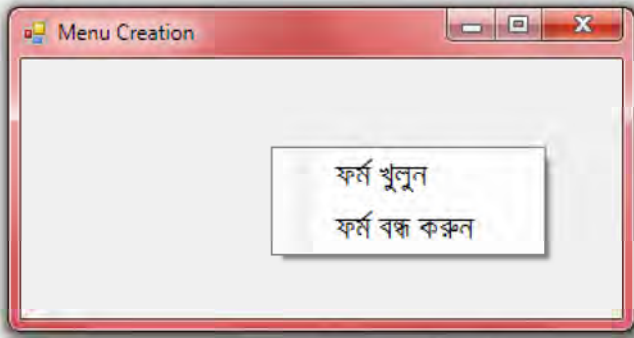
If Form2.Visible = True Then
    Form2.Close()
End If
End Sub

```

প্রোগ্রাম রান করুন: পরের পৃষ্ঠার প্রথম [বামের] চিত্রে রানকালীন অবস্থায় প্রথম ফর্মে রাইট-ক্লিক করার পর কনটেক্সট মেনু আসতে দেখাচ্ছে। ডানের চিত্রে কনটেক্সট মেনু আইটেম থেকে ফর্ম খুলার লাইনটি চয়েজ করায় ফর্মটি খুলেছে।

১৭তম পরিচ্ছেদ: Windows Form Controls

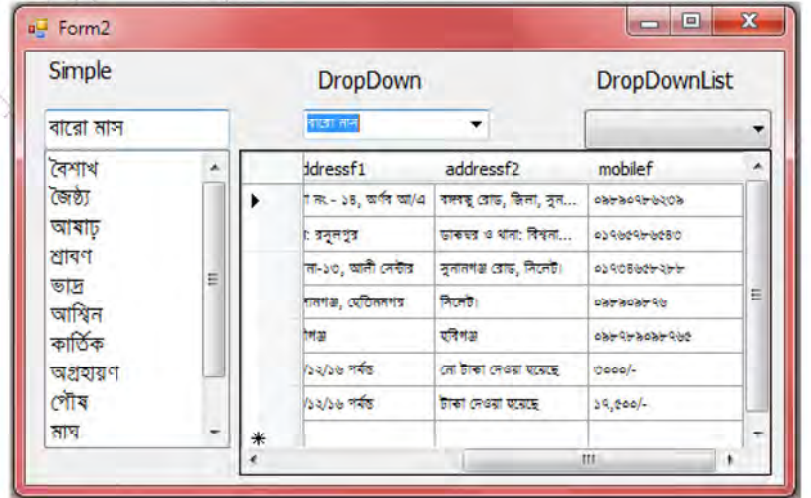
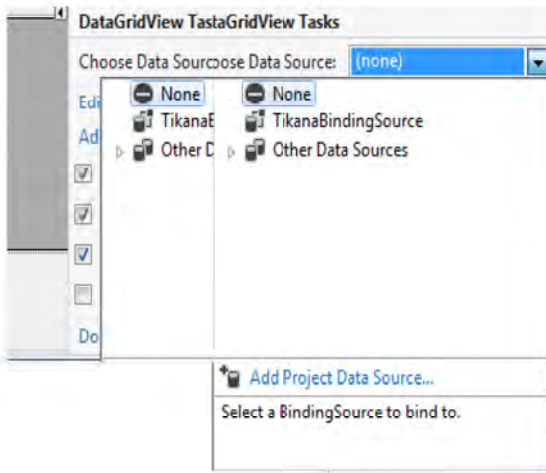
উইন্ডোজ ফর্ম কন্ট্রোলস: Windows Form Controls



11.

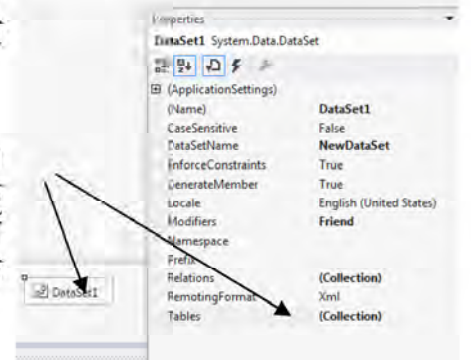
DataGridView: এটা ডাটাবেইজ প্রদর্শনের গ্রিড যা দেখতে অনেকটাই এক্সেস ডাটাবেইজের টেবিলের মতো। একটি উদাহরণ দেখে নিই।

মনে আছে কি- আমাদের ঠিকানা ডাটাবেইজের কথা? গ্রন্থের ১৪০ পৃষ্ঠায় চিত্রসহ নির্দেশনা দেওয়া আছে কিভাবে ডাটাবেইজের সঙ্গে যোগাযোগ সৃষ্টি করা যায়। এখানে অতিরিক্ত যা তাহলো, ডাটাগ্রিডভিউ ফর্মে নিয়ে আসার সাথে সাথেই একটি জানালা আসবে (নিচের বায়ের চিত্র)। যে কন্ট্রোলে লিখা Add Project Data Source... আছে সেখানে যেয়ে নতুন ডাটাবেইজ আনার জন্য ক্লিক করুন। এরপর ১৪০ পৃষ্ঠায় যেভাবে নির্দেশিত হয়েছে সেভাবে ডাটাবেইজের সঙ্গে সংযোগ সৃষ্টি করে প্রোগ্রাম রান করুন [নিচের ডানের বায়ের চিত্র দ্রঃ]।



12. DataSet: এটা হলো ডাটাবেইজ 'মেমোরির মধ্যে উপস্থাপন'। এতে থাকতে পারে ডাটাটেবিল, ডাটারো ও ডাটাকলাম ইত্যাদি। ডাটাসেট ফর্মে নিয়ে আসলে তা নিচের ধূসর এলাকায় দেখা যাবে। এর প্রোপার্টি জানালায় যেয়ে বিভিন্ন ধরনের সম্পর্ক সৃষ্টি করতে পারেন। যেমন আপনি যদি চান ডাটাসেট মেমোরিতে একটি এক্সেস টেবিল রাখবেন, তাহলে আপনি এর প্রোপার্টিতে যেয়ে Tables -এ ক্লিক করলে ডানে কালেকশন শব্দ ও একটি বোতাম দেখতে পাবেন। এতে ক্লিক করলেই টেবিলের সঙ্গে যোগাযোগ তৈরি করতে পারেন।

13. DateTimePicker: এই কন্ট্রোল দিয়ে তারিখ ও সময় প্রদর্শন করা যায়। কন্ট্রোল প্রোপার্টিতে যেয়ে যেভাবে চান সেভাবে প্রদর্শনও সেট করতে পারেন। পরের পৃষ্ঠায় একটি চিত্রে আমরা এর ব্যবহার দেখিয়েছি। একটি ফর্মে এই কন্ট্রোল যুক্ত করে তার বিভিন্ন প্রোপার্টি কিভাবে কাজ করে তা দেখে নিতে পারেন।



১৭তম পরিচ্ছেদ: *Windows Form Controls*

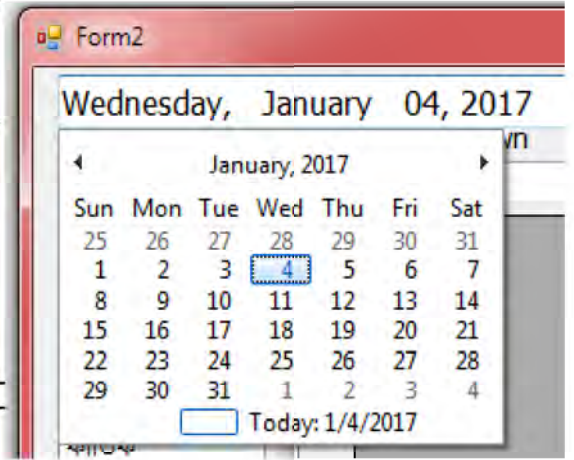
উইন্ডোজ ফর্ম কন্ট্রোলস: *Windows Form Controls*

ডানের চিত্রে ডেইটটাইমপিকার দ্বারা ‘আজকের তারিখ’ একটি ক্যালেন্ডারে কিভাবে দেখানো হয় তা তুলে ধরেছি।

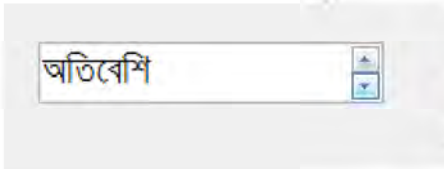
14. DirectoryEntry: এটার কাজ হলো ডাইরেক্টরির ভেতর কি হচ্ছে তা জানা। কোডের মাধ্যমে একে ব্যবহার করে যায়।

15. DirectorySearcher: এটার কাজ হলো ডমেইন সার্ভিস হাইআর্কিতে অনুসন্ধান চালানো।

16. DomainUpDown: এটা অনেকটা **CheckedListBox** এর মতো কাজ করে। এই কন্ট্রলের ভেতর একটি লিস্ট থাকতে পারে। ডানে [আপ-ডাউন এ্যারোতে ক্লিক করে করে লিস্ট দেখা যায়। নিচের চিত্রটি দেখুন।



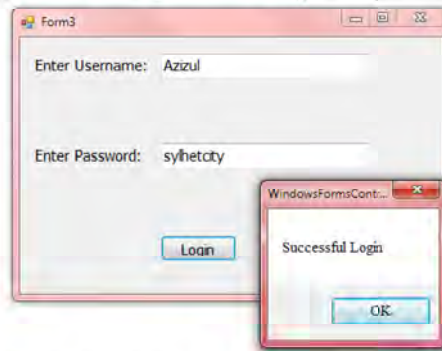
অবশ্য কোডের মাধ্যমে জানা ও সেট করা যায় লিস্টের কোন্ আইটেম চয়েজ করা হয়েছে বা হবে। যেমন:



`DomainUpDown1.SelectedIndex = 0`

17. ErrorProvider: এটা একটি সুবিধা। অর্থাৎ কোনো আইটেম ব্যবহারকারী প্রবেশ করলেন, আর তা আপনি যেভাবে চান তা হয় নি। এটা জানানোর জন্য আইকন ও মেসেজ ব্যবহারকারীকে দেখাতে পারেন। একটি উদাহরণ দ্বারা ব্যাপারটি স্পষ্ট হবে।

মনে করুন আপনি একটি লগিন ফর্ম বানাবেন। ব্যবহারকারী দু’টি ফিল্ডে ইউজারনেইম ও পাসওয়ার্ড প্রবেশ করবেন। আপনি একটি **ErrorProvider** দ্বারা লগিনের ভেলিডিটি চেক করতে পারেন। দেখুন নিচের চিত্র দু’টো ও পরের কোডগুলো।



```
Private Sub btLogin_Click(sender As Object, e As EventArgs) Handles btLogin.Click
    If TextBox1.Text <> "Azzul" Then
        ErrorProvider1.SetError(TextBox1, "Invalid username.")
        Exit Sub
    Else
        ErrorProvider1.SetError(TextBox1, "")
    End If
    If TextBox2.Text <> "sylhetcity" Then
        ErrorProvider1.SetError(TextBox2, "Invalid password.")
        Exit Sub
    Else
        ErrorProvider1.SetError(TextBox2, "")
    End If
    MsgBox("Successful Login")
End Sub
```

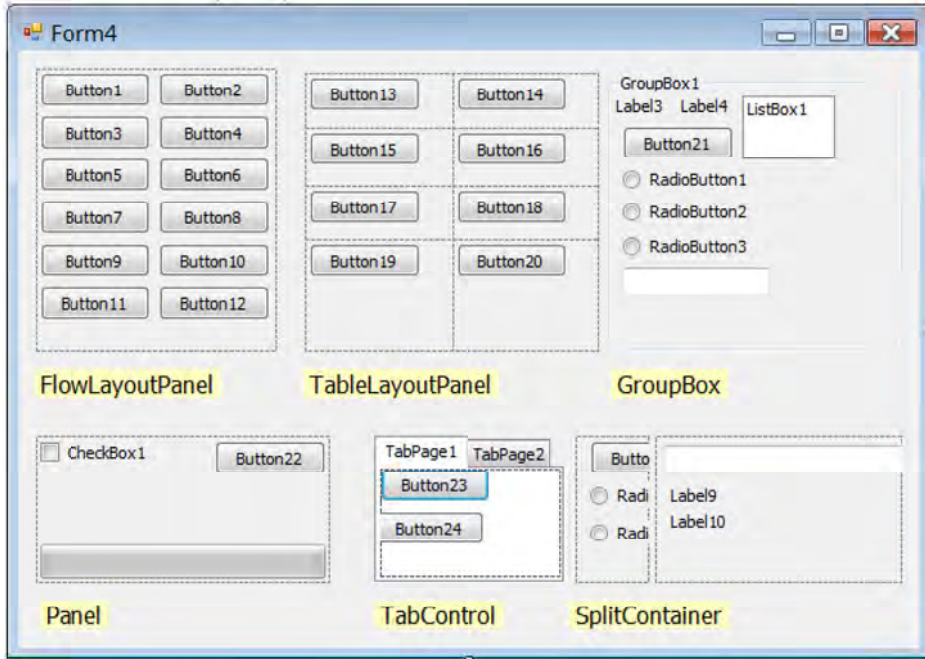
১৭তম পরিচ্ছেদ: *Windows Form Controls*

উইন্ডোজ ফর্ম কন্ট্রোলস: *Windows Form Controls*

18. EventLog: এটার মাধ্যমে কোনো এ্যাপলিকেশনের ইভেন্ট লগ [ঘটনাপ্রবাহ রেকর্ড] তৈরি করা যায়। এটা আমরা খুব অল্প ক্ষেত্রেই ব্যবহার করে থাকি। সুতরাং এ সম্পর্কে বেশি বলার নেই।

19. FileSystemWatcher: এটার কাজ হলো ফাইল সিস্টেমে কী রদবদল হচ্ছে তা জানা ও প্রয়োজনে রিপোর্ট করা। এটা একটি এডভান্স স্তরের ক্রিয়া তাই এটা ব্যবহার পদ্ধতি এখানে দেখানোর প্রয়োজন নেই।

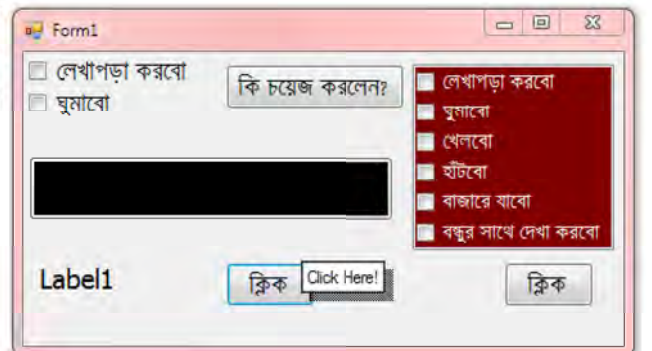
20. FlowLayoutPanel: -এটা Panel জাতীয় কয়েকটি কন্ট্রলের একটি। অন্যান্যগুলো হলো: TableLayoutPanel, GroupBox, TabControl, SplitContainer এবং Panel। এগুলো মূলত ডিজাইনকালে সুবিধার জন্য ব্যবহৃত হয়। নিচের চিত্রটি দেখুন।



22. HelpProvider: যে কোন কন্ট্রলের জন্য সাহায্য নিশ্চিত করার উপায় হিসাবে এই কন্ট্রোলটি ব্যবহৃত হয়ে থাকে। যেমন মনে করুন, আপনি চাচ্ছেন একটি বোতামের একশন কি হবে তা ব্যবহারকারীকে জানাবেন। আপনি চাইলে টুলটিপ ব্যবহার করতে পারেন। তবে এরচেয়েও উন্নতমানের পথ হলো হেল্পপ্রভাইডার। এটির মাধ্যমে শুধুমাত্র সামান্য কিছু লেখা নয়- আরো একটি সংরক্ষিত হেল্প ফাইলও দেখাতে পারেন। ব্যবহারকারী কন্ট্রোলটি ফোকাসে আসার পর কিবোর্ডের "F1" প্রেস করলেই হেল্পপ্রভাইডার সক্রিয় হয়ে ওঠবে। আসুন, একটি উদাহরণ দেখে নিই- বিষয়টি এতে স্পষ্ট হয়ে যাবে। আমরা ফর্মের লোড ইভেন্ট সাবে এই কোডটুকু লিখবো:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    HelpProvider1.SetHelpString(Button3, "Click Here!")
End Sub
```

অবশ্য ফর্মের গ্রে এলাকায় একটি HelpProvider1 আইটেম তো আনতে হবে। টুলবাক্সে যেয়ে এই আইটেমে ডবলক্লিক করলেই কাজটি সারবে। রান করার পর বোতামটি ফোকাসে এলে 'F1' কী চাপুন। ডানের চিত্র দেখুন।



১৭তম পরিচ্ছেদ: *Windows Form Controls*

উইন্ডোজ ফর্ম কন্ট্রোলস: *Windows Form Controls*

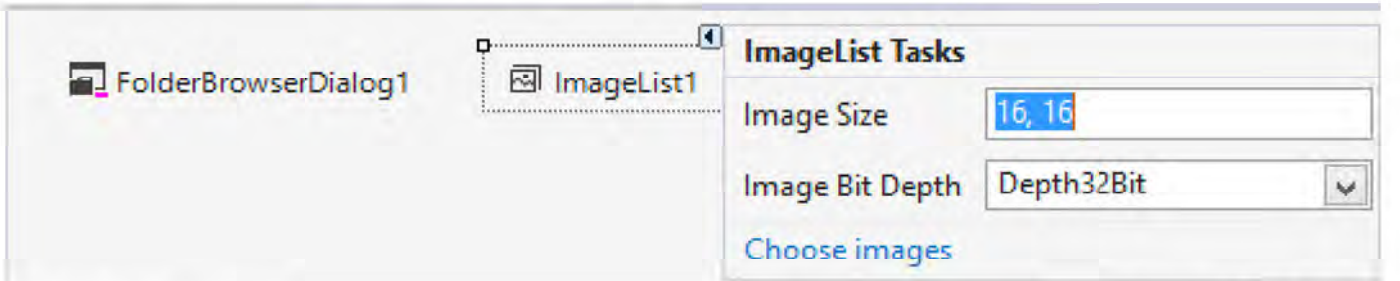
23. HScrollBar: এটা ও 'VScrollBar' কন্ট্রোলের কাজ হলো যথাক্রমে ব্যবহারকারী একটি অংশে ধরে অনুভূমিক ও উপর-নিচে স্ক্রল করতে পারবেন। আমরা একটি উদাহরণ দ্বারা এগুলো বুঝিয়ে দিচ্ছি।



বায়ের চিত্রে আমরা দু'টি স্ক্রল আইটেম দেখছি। এগুলোর মাধ্যমে লেবেল দু'টোর সংখ্যা রদবদল করা যায়। আমরা ডিজাইন টাইমে সর্বোচ্চ ও সর্বনিম্ন সংখ্যা নির্ধারণ করে দিতে পারি। কোন্ কোন্ প্রপার্টি পরিবর্তন করে তা সম্ভব নিচের চিত্রে দেখিয়েছি।

Enabled	True
GenerateMember	True
LargeChange	1
Location	107, 33
Locked	False
Margin	3, 3, 0, 0
Maximum	2017
Minimum	1917
MinimumSize	0, 0

24. ImageList: এটা ছবির একটি লিস্ট স্টোর করে যাতেকরে অন্যান্য কন্ট্রোল বা কোড এগুলো কাজে লাগাতে পারে। আপনি প্রথমে ফর্মে একটি ইমেজলিস্ট নিয়ে আসবেন টুলবক্স থেকে। দেখবেন, এর ডানে একটি ছোট্ট এ্যারো আছে। সিলেক্ট করলেই ইমেজলিস্ট ট্যাস্কস লিস্ট আসবে। আপনি সেখানে ইমেজ সম্পর্কে তথ্যাদি লিখে রাখতে পারেন। ইমেজ লিস্ট নিয়েও আসতে পারেন। দেখুন নিচের চিত্রটি।



25. Label: এটা সম্পর্কে আর কী বলবো। আমরা তো সব সময় এটা ব্যবহার করে আসছি। ব্যবহারকারীকে দৃশ্যত কিছু তথ্য প্রদানের জন্য এটি কাজে লাগানো হয়। অবশ্য রানটাইমে কোডের মাধ্যমে এর টেক্সট, ভিজিবল ইত্যাদি প্রোপার্টি পরিবর্তনও করা যায়।

26. LinkLabel: এটাও মূলত একটি লেবেল। তবে ওয়েভ পেইজে ন্যাভিগেশনের জন্য এটি সহজে ব্যবহার করা যায়। বা ফর্মের ভেতরেও অন্যান্য কাজের জন্য এটি ব্যবহারযোগ্য। এর লেখা থাকে নীল রঙের। এছাড়া আন্ডারলাইনও থাকে। এর উপরে মাউস গেলে হাত আইকনে পরিণত হয়। দেখুন নীচের চিত্রটি। লিংকলেবেলের বেশ কয়েকটি প্রোপার্টি আছে। এগুলো রদবদল করে দেখতে পারেন। একটি ওয়েবসাইটে সরাসরি ভিজিটের জন্য ফর্মের লিংকলেবেলে নিচের কোডটুকু লিখুন।

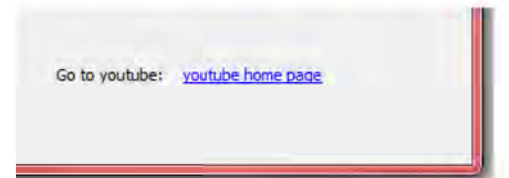
Private Sub LinkLabel1_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs) **Handles** LinkLabel1.LinkClicked

'open the website URL set to the Tag property

' (www.youtube.com)

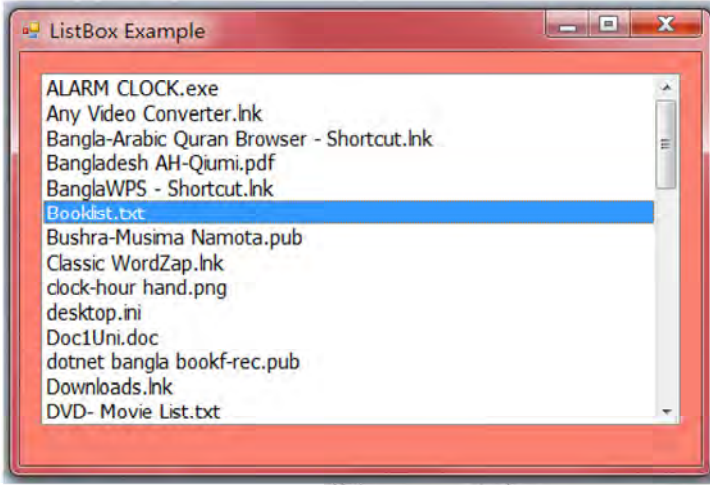
Process.Start(LinkLabel1.Tag.ToString)

End Sub



১৭তম পরিচ্ছেদ: *Windows Form Controls*

উইন্ডোজ ফর্ম কন্ট্রোলস: *Windows Form Controls*



27. ListBox: এটাও আমরা ইতোমধ্যে ব্যবহার করেছি। এতে থাকে নিয়ন্ত্রণযোগ্য কিছু লিস্ট আইটেম। লিস্ট হবে ০ ইনডেক্স। অর্থাৎ চারটি আইটেম থাকলে ইনডেক্স হবে ০,১,২ ও ৩। ইনডেক্স দ্বারা, কিংবা লাইনের সিলেকশন টেক্সট দ্বারা লিস্ট বাক্সের আইটেম সনাক্ত করা যায়। আমরা একটি উদাহরণ দেখাবো। বায়ের চিত্রে একটি জানালার মধ্যে আমরা ফাইলের একটি লিস্ট দেখছি। বাস্তবে এই জানালায় একটি লিস্ট বাক্স আছে। কোডিং এর মাধ্যমে আমরা কম্পিউটারের ডেস্কটপ ফন্ডারের সকল ফাইলের নাম এই বাক্সে প্রদর্শন করেছি। নিচে এই কোডটুকু দেখুন। একটি ফর্ম আপনার প্রজেক্টে যুক্ত করুন। এতে টুলবাক্স থেকে নিয়ে আসুন একটি লিস্টবক্স। ফর্মের কোথাও ডবল ক্লিক করুন। কোড এডিটরের Form_Load ইভেন্ট সাবে

নিচের কোডটি লিখুন। আপনি চাইলে অন্য যে কোনো ফন্ডারের নামও দিতে পারেন। বরং বিভিন্ন ফন্ডানের নাম দিয়ে টেস্ট করে দেখুন।

Private Sub Form6_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Dim mydoc As String = My.Computer.FileSystem.SpecialDirectories.Desktop

Dim path As String = IO.Path.Combine(mydoc)

Dim dir As New IO.DirectoryInfo(path)

' populate the listbox control

For Each file In dir.GetFiles

' avoid jpg images

If Not file.Extension = ".jpg" Then

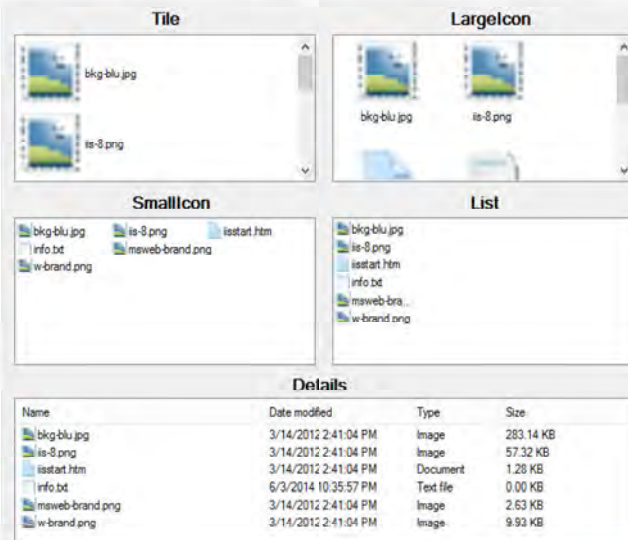
ListBox1.Items.Add(file.Name)

End If

Next

End Sub

28. ListView: এটাতে ফাইলের লিস্ট দেখানো যায়। তবে আরো বেশি তথ্যসহ। মোট ৫ উপায়ে লিস্টভিউ কন্ট্রোলে ফাইল নাম প্রদর্শন সম্ভব। বায়ের চিত্রটি দেখুন।



ক. Tiles: আইটেমের নামের বামপাশে বড়ো বড়ো আইকন দেখায়।

খ. LargeIcon: আইটেমের নামের উপরে বড়ো আইকন দেখায়।

গ. SmallIcon: নামের বায়ে ছোট আইকন প্রদর্শন করে।

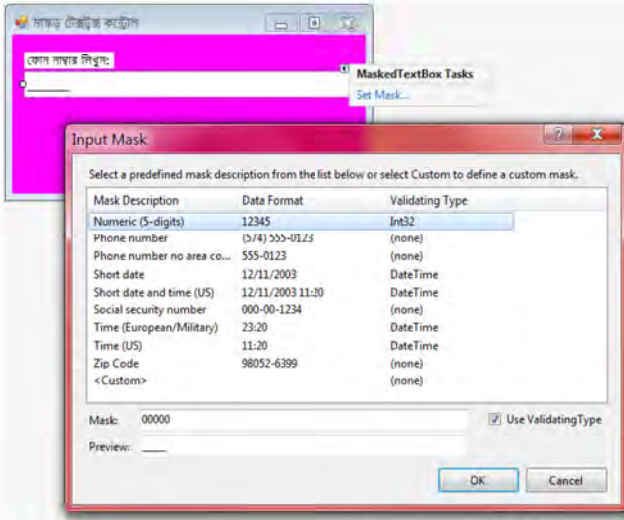
ঘ. List: নামের বায়ে ছোট আইকন দেখায় ও প্রতিটি আইটেম একেকটি লাইনে থাকে।

ঙ. Details: ছোট আইকনসহ ফাইলের পূর্ণাঙ্গ তথ্য প্রদর্শন করে।

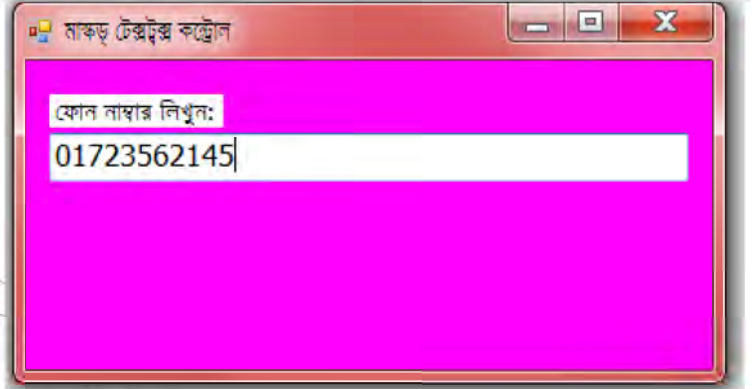
29. MaskedTextBox: এটা টেক্সট বাক্সের মতোই কাজ করে। তবে অতিরিক্ত যা তাহলো ব্যবহারকারী ফিল্ডের মধ্যে কী লিখবেন তা নির্ধারণ করা যায়। এক ধরনের ভেলিডিটি ফিল্ড এটি। উদাহরণ দেখুন।

১৭তম পরিচ্ছেদ: *Windows Form Controls*

উইন্ডোজ ফর্ম কন্ট্রোলস: *Windows Form Controls*

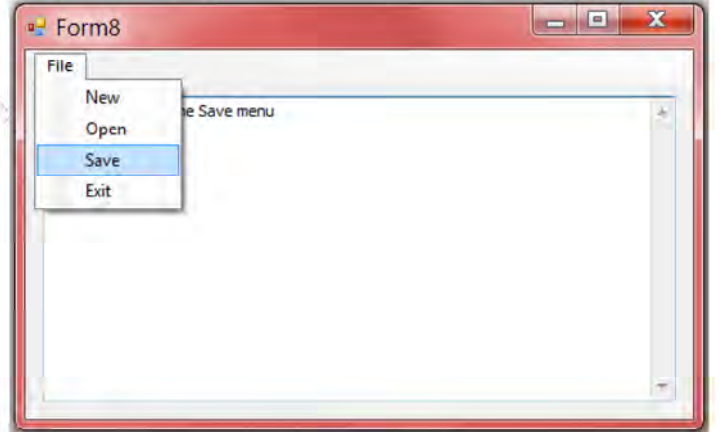
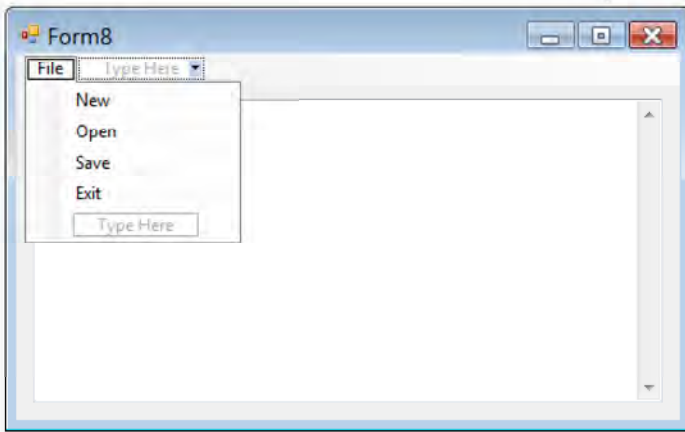


টুল বাক্স থেকে ফর্মে যুক্ত করুন **MaskedTextBox** একটি। টেক্সবক্সের ডানের দিকে উপরের কোণে যে এ্যারো দেখা যাবে তাতে ক্লিক করুন। চয়েজ করুন **Set Mask ...**। বায়ে চিত্রিত জানালাটি আসবে। এখান থেকে যে কোনো একটি চয়েজ করে 'OK' ক্লিক করুন। এবার প্রোগ্রাম রান করুন। দেখবেন আপনার



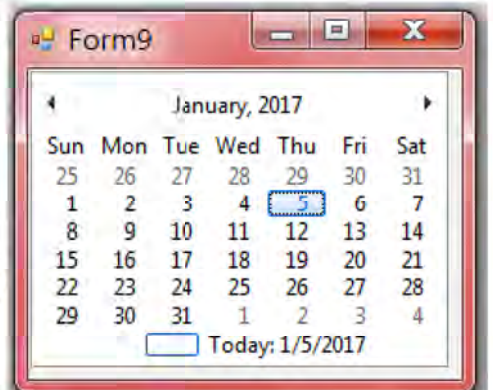
চয়েজকৃত উপায়েই লেখা চুকানো সম্ভব হচ্ছে টেক্সট ফিল্ডে। লক্ষ করুন, ফিল্ডের মধ্যে ১১টি মাত্র সংখ্যা চুকানো সম্ভব হয়েছে।

30. MenuStrip: এ কন্ট্রোলার মাধ্যমে ফর্মের মধ্যে মেনু, সাবমেনু, মেনু আইটেম ইত্যাদি তৈরি করা যায়। প্রত্যেকটির সঙ্গে কোড যুক্ত করে কাজকৃত একশনও আঞ্জাম দেওয়া যায়। নিচের চিত্রগুলো দেখুন।



31. MessageQueue এটা একটি উচ্চ পর্যায়ের ক্লাস। সার্ভারের সাথে মেসেজ যোগাযোগ কিউ নিয়ন্ত্রণে এটি ব্যবহৃত হয়। এ টুলবক্স আইটেম সম্পর্কে আর অতিরিক্ত আলোচনা গ্রন্থের স্কোপের বাইরে। সুতরাং আর কিছু বলার নেই। এ সম্পর্কে অতিরিক্ত জানতে চাইলে ইন্টারনেটের এই লিংক দেখতে পারেন: <http://msdn.microsoft.com/en-us/library/system.messaging.messagequeue.aspx>

32. MonthCalendar বুঝতেই পারছেন, এটা হলো পুরো মাসের ক্যালেন্ডার কন্ট্রোল। ক্যালেন্ডারের রেঞ্জ বিরাট: ১৭৫৩ থেকে ৯৯৯৮ সাল পর্যন্ত এ রেঞ্জের ব্যাপ্তি। আপনি এরই ভেতর যে কোন তারিখ যথাক্রমে সর্বনিম্ন ও সর্বোচ্চ হিসেবে প্রোপার্টিতে যেয়ে সেট করতে পারেন। ডানের চিত্রটি দেখুন।



33. NotifyIcon: এটা প্রোগ্রাম রানকালে অদৃশ্য থাকে। একে ব্যবহার করা

১৭তম পরিচ্ছেদ: *Windows Form Controls*

উইন্ডোজ ফর্ম কন্ট্রোলস: *Windows Form Controls*

যায় সিস্টেম ট্রেতে আইকোন প্রদর্শন করে।



34. NumericUpDown: এ কন্ট্রোল একটি সংখ্যা প্রদর্শন করে। এতে থাকে একটি আপ-ডাউন এ্যারো। আপনি প্রোপার্টিতে যোগে সর্বোচ্চ ও সর্বনিম্ন সংখ্যা, কত করে বাড়বে বা কমবে ক'টি ডেসিমেল স্থান হবে ইত্যাদি সেট করতে পারেন। কন্ট্রোলার ValueChanged ইভেন্ট সক্রিয় হয় যখনই ব্যবহারকারী কিংবা কোড এর মান পরিবর্তন করে। এই ইভেন্ট সাবকে ধরে কোডিং করা যায়। দেখুন বায়ের চিত্রটি।

35. OpenFileDialog -এটার মাধ্যমে ফাইল খোলার ডায়ালগ প্রদর্শিত হয়। এটা একটি শক্তিশালী কন্ট্রোল। অবশ্য ভিজুয়াল বেসিক.নেট বেশ ক'টি অনুরূপ ডায়ালগ সৃষ্টি করে দিয়েছে। অপেন ডায়ালগের কাজ হলো কম্পিউটার থেকে কোনো ফাইল সিলেক্ট করে খোলার জন্য প্রস্তুত করা। ডায়ালগের ফলাফল হয় 'OK' না হয় 'Cancel'। এই কন্ট্রোলে বেশ ক'টি প্রোপার্টি আছে। এগুলো রানটাইমে ব্যবহার করা যায়। যেমন Filter প্রোপার্টির মাধ্যমে নিশ্চিত করতে পারেন যে ব্যবহারকারী শুধুমাত্র .jpg ও .png ফাইল টাইপ খোলা ছাড়া অন্য কিছু দেখতেই পারেন না। আমরা কোডে লিখতে পারি: OpenFileDialog1.Filter = "Jpeg Files (*.jpg)|*.jpg|Png Files (*.png)|*.png"

যে ফাইল সিলেক্ট করা হয়েছে তার নাম জানতে হলে আমরা ব্যবহার করতে পারি FileName প্রোপার্টি, যেমন:

```
Private Sub BrowseButton_Click(sender As Object, e As EventArgs) Handles BrowseButton.Click
    If OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
        SelectedFile.Text = OpenFileDialog1.FileName
    End If
End Sub
```

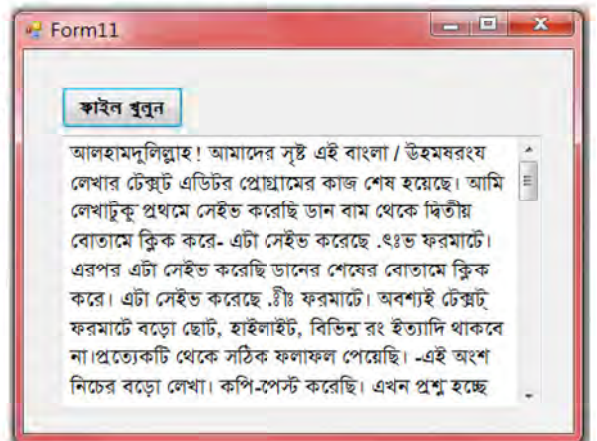
আরো একটি উল্লেখযোগ্য প্রোপার্টি হচ্ছে MultiSelect। এটা দ্বারা সিদ্ধান্ত নেওয়া যায় ব্যবহারকারী একাধিক ফাইল সিলেক্ট করতে পারবে কি না। যেমন:

```
Private Sub BrowseButton_Click(sender As Object, e As EventArgs) Handles BrowseButton.Click
    If OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
        For Each file As String In OpenFileDialog1.FileNames
            MessageBox.Show(file)
        Next
    End If
End Sub
```

মেসেজ বাক্সে একে একে সকল ফাইলের নাম প্রদর্শিত হবে। এবার আমরা ছোট্ট একটি উদাহরণ দেখে নেবো।

একটি নতুন ফর্ম প্রজেক্ট তৈরি করুন। যোগ করুন একটি বোতাম, একটি টেক্সট বাক্স ও একটি অপেন ফাইল ডায়ালগ কন্ট্রোল। বোতামে ডবল ক্লিক করুন। কোড জানালাতে ফর্মের ক্লাসের ভেতর যে ইভেন্ট সাব আসবে তার ভেতর লিখুন:

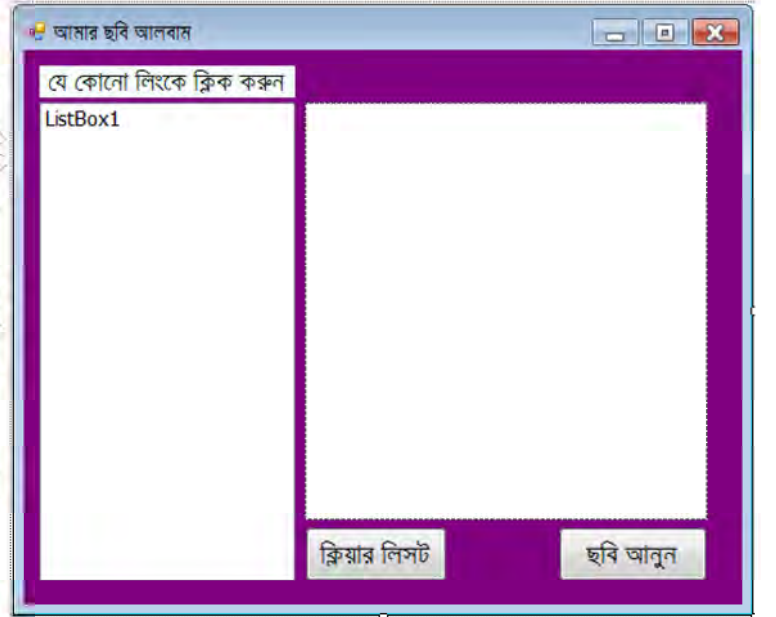
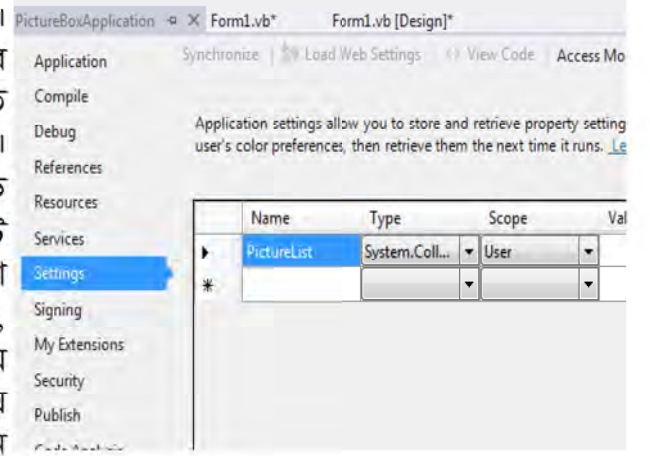
```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        ' Displays an OpenFileDialog so the user can select a Cursor.
        Dim openFileDialog1 As New OpenFileDialog()
```



১৭তম পরিচ্ছেদ: *Windows Form Controls*

উইন্ডোজ ফর্ম কন্ট্রোলস: *Windows Form Controls*

39. PictureBox: এই কন্ট্রলের মাধ্যমে ছবি প্রদর্শন করা যায়। এছাড়া এটি একটি গ্রাফিক্স অবজেক্ট হিসেবেও কাজ করে যার মধ্যে আপনি বিভিন্ন ধরনের আকার রানটাইমে অঙ্কন করতে পারেন। ডিজাইন টাইমে দু'টি প্রোপার্টি খুব ব্যবহারোপযোগী। এগুলো হলো তার ইমেজ [Image] এবং সাইজমোড [SizeMode] প্রোপার্টি। প্রথমটি দ্বারা যে কোনো ছবি ডিফোল্ট হিসাবে প্রদর্শন ও দ্বিতীয়টি দ্বারা ছবিটি কিভাবে প্রদর্শিত হবে তা নির্দিষ্ট করা যায়। ছবিটি কন্ট্রলের পুরো এলাকা জুড়ে, মধ্যখানে, বড় হয়ে ইত্যাদি কোন উপায়ে প্রদর্শিত হবে- তা জাইনটাইমে করার সুযোগ সৃষ্টি করে দিয়েছে দ্বিতীয় প্রোপার্টিটি। পিকচার বাস্ক সম্পর্কে আরো কিছু জানতে আমরা একটি সুন্দর এ্যাপলিকেশন তৈরি করবো। এই এ্যাপলিকেশন হবে ছবি আলবামের মতো। আপনি যতোটা প্রয়োজন ছবি কম্পিউটারের যে কোনো ফন্ডার থেকে এনে পিকচার বাস্কে প্রদর্শন করতে পারেন। আরো যা পারবেন তাহলো পরবর্তিতে প্রোগ্রাম রান করলে সকল ছবির লিংক একটি লিস্ট বাস্কে দেখতে পাবেন। শুধু মাত্র যে কোনো লিংকে ক্লিক করলেই ছবিটি ভেসে আসবে পিকচার বাস্কে। সুতরাং আপনার ছবি আলবাম বানাতে নিচের পদক্ষেপগুলো অনুসরণ করুন।



১. একটি ফর্ম প্রজেক্ট তৈরি করুন। এর নামকরণ করুন: MyPictureAlbum।

২. ফর্মে টুলবাক্স থেকে নিয়ে আসুন দু'টি বোতাম, একটি পিকচার বাস্ক, একটি লেবেল, একটি লিস্ট বাস্ক ও একটি অপেন ফাইল ডায়ালগ। এরপর ইচ্ছেমতো ডিজাইন করুন। দেখুন আমার ডিজাইনটি।

৩. Solution Explorer এ য়েয়ে ডবল ক্লিক করুন, My Project। একটি জানালা আসবে। নিচে এর একটি অংশ দেখানো হলো।

৪. সিলেক্ট করুন, Settings। Name কলামে টাইপ করুন: PictureList। পাশের Type কলামের এ্যারোতে

যেয়ে সিলেক্ট করুন: System.Collections.Specialized.StringCollection। ব্যস! এটি বন্ধ করুন। এই সেটিংয়ের কারণে আমরা লিস্ট বাস্কের আইটেমগুলো সংরক্ষণ করতে পারবো।

৫. এবার কিছু কোডিং লাগবে। (ক) 'ছবি আনুন' বোতামে ডবল ক্লিক করে কোড এডিটর খুলুন। বোতামের ক্লিক ইভেন্টে লিখুন:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim result As DialogResult
    'set filter
    OpenFileDialog1.Filter = "JPEG Files (*.jpg)|*.jpg|Bitmap Files (*.bmp)|*.bmp| Gif files (*.gif)|*.gif| Png files (*.png)|*.png"
    'Enable multi-file names selection
    OpenFileDialog1.Multiselect = True
    'Open file dialog
    result = OpenFileDialog1.ShowDialog
    If result = DialogResult.OK Then
```

১৭তম পরিচ্ছেদ: *Windows Form Controls*

উইন্ডোজ ফর্ম কন্ট্রোলস: *Windows Form Controls*

```
For Each filename In OpenFileDialog1.FileNames
    'add selected file names to listbox
    ListBox1.Items.Add(filename)
Next
End If
End Sub
```

এই কোডের কাজ হলো OpenFileDialog এর মাধ্যমে এক বা একাধিক ছবির পূর্ণাঙ্গ প্যাথ স্ট্রিং লিস্টবাক্সে নিয়ে আসা। প্রথম লাইন দ্বারা আমরা ডায়ালগ থেকে যে ফলাফল আসবে তার জন্য একটি ডিক্লারেশন করেছি। দ্বিতীয় লাইন দ্বারা নির্দিষ্ট করেছি কোন্ কোন্ ধরনের ইমেজ ফাইল আনা যাবে। এর পরের লাইনে অপেন ডাইলগের Multiselect নামক প্রোপার্টি সত্য হিসাবে সেট করেছি। এর ফলে একই সঙ্গে একাধিক ছবি সিলেক্ট করতে পারবেন ব্যবহারকারী। এর পরের লাইনে ব্যবহারকারী ক্যানসেল না করলে আমরা প্রতিটি সিলেক্টেড ছবির ফাইল নাম [তাদের প্যাথ নেইম] লিস্ট বাক্সে নিয়ে এসেছি। অবশ্য ইতোমধ্যে লিস্ট বাক্সে কিছু থাকলে তা মুছে যাবে না।

৬. লিস্ট বাক্সে ডবল ক্লিক করুন। যে সাব আসবে তাতে লিখুন এই কোডটুকু:

```
Private Sub ListBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    ListBox1.SelectedIndexChanged
    'View image in picturebox when a file name was selected from the list
    PictureBox1.Image = Image.FromFile(ListBox1.SelectedItem.ToString)
End Sub
```

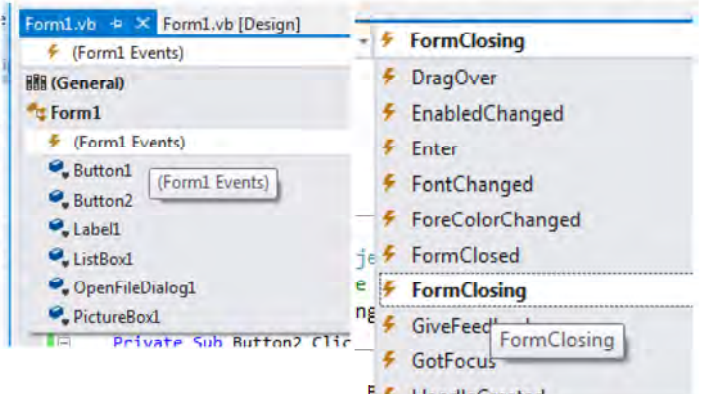
এখানে আমরা একটি মাত্র লাইন লিখেছি। লিস্ট বাক্সে যে ইনডেন্টে ক্লিক করা আছে সে প্যাথ নাম অনুযায়ী পিকচার বাক্সে ছবি এনে প্রদর্শনের নির্দেশ দেওয়া হয়েছে এ লাইনের মাধ্যমে।

৭. এবার ডবল ক্লিক করুন ক্লিয়ার লিস্ট বোতামে। ক্লিক ইভেন্ট সাবে লিখুন:

```
'clear list button code
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    PictureBox1.Image = Nothing
    Dim Res As Integer = MsgBox("সকল প্যাথ নাম মুছে দেবেন?", MsgBoxStyle.YesNo)
    'MsgBox(Res)
    If Res = 6 Then
        ListBox1.Items.Clear()
    End If
End Sub
```

এই সাবের প্রথম লাইন দ্বারা প্রদর্শিত ইমেজ (পিকচার বাক্সে থাকলে) মুছে দিয়েছি। এরপর একটি ওয়ার্নিং দ্বারা ব্যবহারকারীকে জানানো হচ্ছে সত্যিই কি লিস্টটি মুছে দেবেন? ফলাফল সত্যি হলে [৬] পরের লাইন নির্দেশ দেওয়া হয়েছে লিস্ট বাক্সের সবকিছু মুছে ফেলে দিতে।

৮. Form Load এবং Form Closing ইভেন্ট সাবের কোড: আমরা এ দুটো ইভেন্ট ধরে কিছু কাজ করেছি। লোড হওয়ার সময় আমরা চাই সিস্টেম সেটিংয়ে সংরক্ষিত লিস্টটি লিস্ট বাক্সে নিয়ে আসতে। আর বন্ধ হওয়ার সময় আমরা চাই লিস্ট বাক্সের আইটেমগুলো আবার সিস্টেম সেটিংয়ে সংরক্ষণ করতে পারি। ইতোমধ্যে ব্যবহারকারী লিস্টে পরিবর্তন আনতে পারেন। প্রথম সাবরুটিন আসবে ফর্মের টাইলেট বারে ডবল ক্লিক করলে। কিন্তু দ্বিতীয়টি ফর্মের ইভেন্ট লিস্ট থেকে আনতে হবে। দেখুন ডানের চিত্রটি। এটা আনার ক্লিক পরম্পরা: ক্লিক Form1.vb > Form1 > (Form1 Event > FormClosing। ব্যস! Form1_FormClosing নামক একটি ইভেন্ট হ্যান্ডলার আসবে।



১৭তম পরিচ্ছেদ: *Windows Form Controls*

উইন্ডোজ ফর্ম কন্ট্রোলস: *Windows Form Controls*

যাক, প্রথমে আমরা ফর্ম লোডের কোডের ওপর ব্যাখ্যা দেবো। নিচে এই কোডগুলো দেখানো হয়েছে।

'get whatever in settings to the ListBox

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles Me.Load
    For Each item As String In My.Settings.PictureList
        ListBox1.Items.Add(item) 'my.settings items adding to listBox1
    Next
End Sub
```

লক্ষ করুন, পিকচার লিস্টে যে ক'পি রেকর্ড আছে তা আমরা একে একে লিস্ট বাক্সে এনে প্রদর্শন করেছি। সুতরাং আলবামের সকল সংরক্ষিত ছবি পিকচার বাক্সে আনতে কোনো বাধা নেই। শুধুমাত্র লিংকে ক্লিক করলেই কাজ হবে। এবার দেখে নেবো ফর্ম ক্লোজিং সাবের কোড।

'save whatever in the list to setting as lines of strings

```
Private Sub Form1_FormClosing(sender As Object, e As FormClosingEventArgs) Handles Me.FormClosing
    My.Settings.PictureList = New Specialized.StringCollection() 'pictureList collection Instant declaration
    For Each item As String In ListBox1.Items
        My.Settings.PictureList.Add(item) 'ListBox items
    Next
    My.Settings.Save() 'and save it
Next
End Sub
```



এ কোডগুলোর পথম লাইন সেটিংয়ে সংরক্ষিত লিস্ট মুছে দেবে ও একটি নতুন কালেকশন তৈরি করবে। এরপর লিস্ট বাক্সের প্রতিটি লাইন একে একে ঐ কালেকশনে নিয়ে সংরক্ষণ করবে। সুতরাং আলবামের সকল লিংক সংরক্ষিত হবে। ব্যবহারকারী যখন প্রোগ্রাম আবার রান করবেন তখন ঐ কালেকশন থেকে লিংকগুলো উপরের ফর্ম লোড ইভেন্ট সাবের কোডের মাধ্যমে লিস্ট বাক্সে চলে আসবে। সুতরাং আমরা সত্যিকার অর্থে একটি ছবি আলবাম তৈরি করলাম। বায়ে এর রানটাইমকালীন একটি চিত্র দিয়েছি। দেখে নিন।

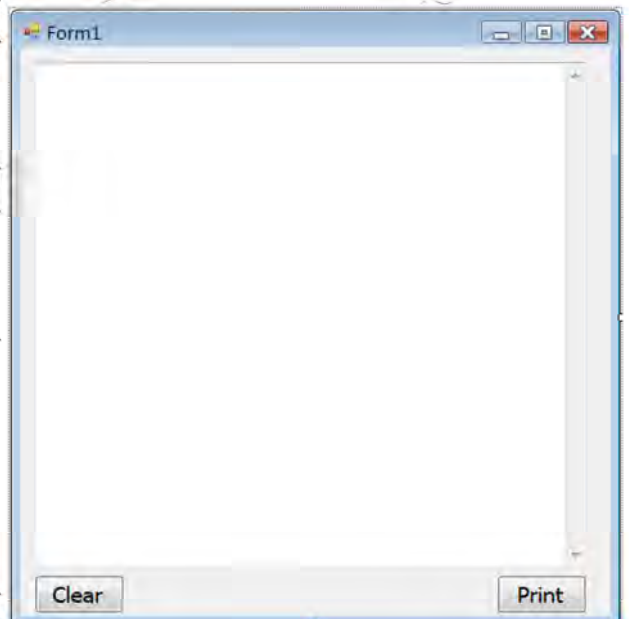
40. PrintDialog: এই ডায়ালগ ও PrintDocument-অবজেক্ট একত্রে কাজ করে আপনার টেক্সট প্রিন্ট করতে পারেন ডিফোল্ট প্রিন্টারে। আমরা একটি উদাহরণ তুলে ধরছি।

একটি নতুন প্রজেক্ট তৈরি করুন। ফর্মের মধ্যে যুক্ত করুন দু'টি বোতাম ও একটি টেক্সট বাক্স। এরপর টুলবাক্স থেকে ডবল ক্লিক করে সলুশনে নিয়ে আসুন **PrintDialog** এবং **PrintDocument** অবজেক্ট। ডানের চিত্রটি দেখুন।

এবার ক্লিয়ার বোতামে ডবল ক্লিক করুন এবং সাবের ভেতর নিচের কোড লাইনটি লিখুন।

```
Private Sub Button2_Click(sender As Object, e As EventArgs)
    Handles Button2.Click
    TextBox1.Text = ""
End Sub
```

এই কোড টেক্সট বাক্সকে খালি করে দেবে। এবার ডবল ক্লিক করুন

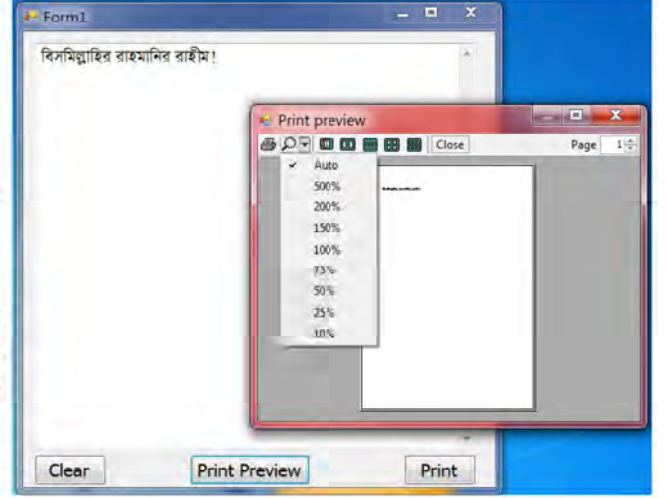


১৭তম পরিচ্ছেদ: *Windows Form Controls*

উইন্ডোজ ফর্ম কন্ট্রোলস: *Windows Form Controls*

```
Else
    PrintPreviewDialog1.Show()
End If
End Sub
```

এই যা- আর কিছু লাগবে না। প্রিন্ট প্রিভিউ করেও এবার প্রিন্টিং করতে পারেন। ডানের চিত্রটি দেখুন।

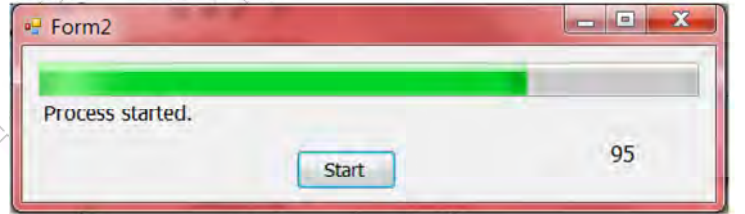


44. Process: এটার কাজ হলো কম্পিউটারে চলন্ত প্রসেসগুলো সম্পর্কে তথ্য সংগ্রহ করা ও প্রয়োজনে প্রদর্শন করা। এটা মূলত সিস্টেম ডায়াগনস্টিক এর একটি ফাংশন। এডভান্স স্তরের প্রোগ্রামাররা একে ব্যবহার করতে পারেন। সুতরাং আমাদের ক্ষেত্রে এর ওপর আলোচনা আদৌ সঠিক হবে না।

45. ProgressBar: ওয়েভ ব্রাউজার উদহরণ প্রোগ্রাম তৈরির সময় আমরা এর ব্যবহার দেখেছি। কোনো দীর্ঘ প্রসেসের সময় ব্যবহারকারীকে ফিডব্যাক দিতে এটি কাজে লাগানো যায়। এর দু'টি গুরুত্বপূর্ণ প্রোপার্টি আছে: Minimum এবং Maximum। প্রথমাবস্থায় বারটি সম্পূর্ণ রংহীন থাকবে আর দ্বিতীয় অবস্থায় সম্পূর্ণরূপে রঙে পরিপূর্ণ হবে। একটি উদাহরণ দেখে নিই।

`Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click`

```
Label1.Text = "Process started."
For x As Integer = 0 To 100
    Label2.Text = x
    ProgressBar1.PerformStep()
    wait(1) 'wait for a millisecond
Next
ProgressBar1.Refresh()
Label1.Text = "Process Ended."
End Sub
```



' Loops for a specified period of time (milliseconds)

```
Private Sub wait(ByVal interval As Integer)
    Dim sw As New Stopwatch
    sw.Start()
    Do While sw.ElapsedMilliseconds < interval
        ' Allows UI to remain responsive
        Application.DoEvents()
    Loop
    sw.Stop()
End Sub
```

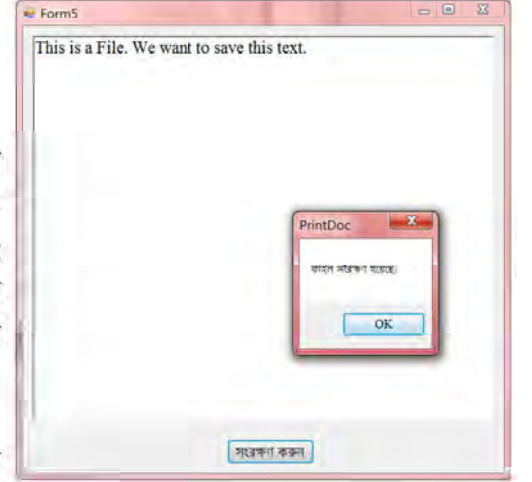
উপরের wait সাবরুটিন দ্বারা নির্দিষ্ট মিলিসেকেন্ড পরিমাণ অপেক্ষার ব্যবস্থা করেছি। এটা প্রগ্রেস বারের মাত্রা বৃদ্ধির পরই বার বার কল করেছি যাতে করে বারের গতিটি দৃশ্যমান হয়।

46. PropertyGrid: এটা আইডিই [Integrated Development Environment] এর মতো বস্তুর প্রোপার্টিগুলো দেখায়। কন্ট্রোলের দু'টি গুরুত্বপূর্ণ প্রোপার্টি হলো: ১. SelectedObject এবং ২. SelectedObjects। প্রথমটির অর্থ একটি অবজেক্ট ও দ্বিতীয়টির অর্থ একাধিক অবজেক্ট। রানটাইমে অবজেক্টের প্রোপার্টি রদবদল করা যায়। আমরা একটি উদাহরণ দেখাবো।

১৭তম পরিচ্ছেদ: *Windows Form Controls*

উইন্ডোজ ফর্ম কন্ট্রোলস: *Windows Form Controls*

```
MsgBox("ফাইল সংরক্ষণ হয়েছে।")
End If
End Sub
```



50. SerialPort: -এটা দ্বারা কম্পিউটারের যে কোনো ফিজিক্যাল সিরিয়াল পোর্টের বিভিন্ন প্রোপার্টি সম্পর্কে অবগত হওয়া যায়। এগুলোর মধ্যে আছে: baud rate, break signal, Data Set Ready (DSR) state, port name, parity, and stop bits ইত্যাদি। নোট করুন, এসব ব্যাপার অনেকটা এডভান্স স্টেজের প্রোগ্রামিংয়ের সঙ্গে জড়িত। সুতরাং এ গ্রন্থে এসব ব্যাপার নিয়ে আলোচনা আদৌ সঠিক হবে না।

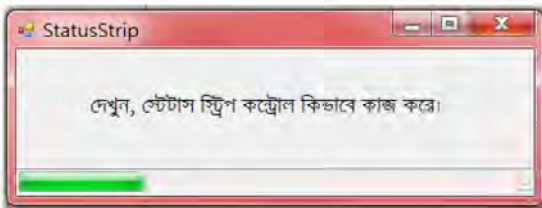
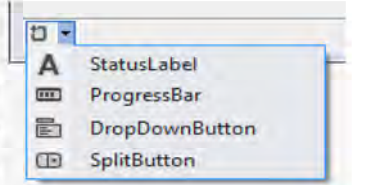
51. ServiceController: এটাও উচ্চ পর্যায়ের প্রোগ্রামারদের সুবিধার্থে তৈরি করেছে মাইক্রোসফট। উইন্ডোজ সার্ভিস প্রসেস ও মেথড এতে জড়িত। আমরা এ সম্পর্কে অতিরিক্ত কিছু বলতে পারবো না।

52. SplitContainer: এ কন্ট্রোল সম্পর্কে কিছুটা ব্যাখ্যা ১৭৭ পৃষ্ঠায় হয়েছে। এটি মূলত ডিজাইন কম্পোনেন্ট। এ সম্পর্কে এখানে আর কিছু বলছি না।

53. Splitter: এটার দ্বারাও ডিজাইনকালে কিছু কাজ করা যায়।

54. StatusStrip: এটা অনেকটা টুলস্ট্রিপের মতো। ফর্মের নিচে এটি প্রদর্শিত হয়। এর উদ্দেশ্য হলো স্টেটাস তথ্যাদি সংক্ষিপ্তভাবে দেখানো। এতে কয়েকটি বস্তু থাকে: স্টেটাস লেবেল, প্রগ্রেস বার, ড্রপডাউনবাটন এবং স্প্লিটবাটন। এগুলোর প্রতিটি আবার ফর্মের একেকটি কন্ট্রোল দ্বারা রেপ্রেজেন্ট করে। যেমন: প্রগ্রেসবারকে রেপ্রেজেন্ট করে টুলস্ট্রিপবার কন্ট্রোল। উদাহরণ থেকে ব্যাপারটি স্পষ্ট হবে। একটি ফর্মে যুক্ত করুন একটি স্টেটাস স্ট্রিপ কন্ট্রোল। ছোট্ট এ্যারোতে যেয়ে প্রথমে চয়েজ করুন: ProgressBar। ফর্মে ডবলক্লিক করে এর কোড এডিটর খুলুন। যুক্ত করুন নিচের কোডগুলো লোড ইভেন্ট সাবে:

```
While ToolStripProgressBar1.Value < 100
    ToolStripProgressBar1.Value += 5
End While
```



রান করলে বায়ের চিত্রটি দেখতে পাবেন।

55. TabControl: এই কন্ট্রোল পৃষ্ঠায় পৃষ্ঠায় গ্রুপকৃত তথ্যাদি প্রদর্শন করে থাকে। আমরা এর ব্যবহার ১৭৭ পৃষ্ঠায় দেখেছি। সুতরাং এখানে আর অতিরিক্ত কিছু বলার নেই- তবে! একটি উদাহরণ দেখে নিই।

একটি নতুন প্রজেক্ট তৈরি করুন। ফর্মের মধ্যে একটি ট্যাব কন্ট্রোল যুক্ত করুন। আমাদের উদাহরণে মোট ৫টি ট্যাব আছে। কন্ট্রোলার ডানের উপরের কোণে যে ছোট্ট এ্যারো আছে তাতে ক্লিক করে Abb Tab দ্বারা যতোটি প্রয়োজন ট্যাব যুক্ত করতে পারেন। প্রত্যেক ট্যাবের আলাদা জানালা আছে যাতে যুক্ত করা যায় অন্যান্য কন্ট্রোল। আমি প্রথম ট্যাবে একটি পিকচার বাক্স ও একটি লেবেল যুক্ত করেছি। পিকচার বাক্সে দেখাবো একটি ছবি এবং লেবেলে এ ছবি সম্পর্কে তথ্য থাকবে। অন্যান্য ট্যাবেও বিভিন্ন বস্তু যুক্ত করেছি। এবার দেখে নিন কোডটুকু।

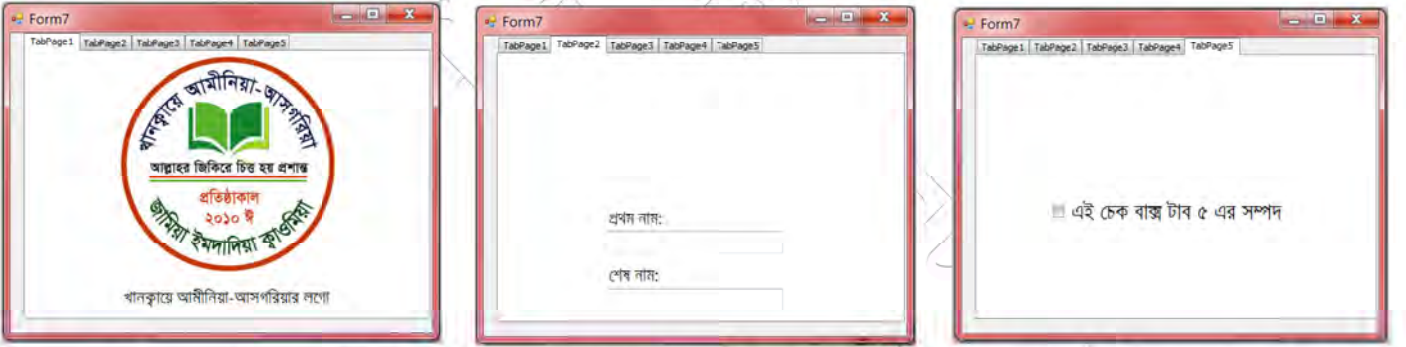
১৭তম পরিচ্ছেদ: *Windows Form Controls*

উইন্ডোজ ফর্ম কন্ট্রোলস: *Windows Form Controls*

```
Private Sub TabControl_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    'select the tabpage #3
    TabControl1.SelectedIndex = 4
End Sub
```

```
Private Sub TabControl1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
    TabControl1.SelectedIndexChanged
    If TabControl1.SelectedIndex = 0 Then
        PictureBox1.Image = My.Resources.LogoA
    End If
End Sub
```

উপরের প্রথম সাবরুটিনে আমরা ট্যাবের ডিফোল্ট ইনডেক্স কি হবে তা নির্দেশ করেছি। আর দ্বিতীয় সাব দ্বারা ইনডেক্স সিলেকশন ০ হলে ছবি প্রদর্শনের কথা বলেছি। ছবিটি অবশ্য My.Resources- এ ইতোমধ্যে এনে রেখেছিলাম। এবার দেখুন রানটাইমের ক'টি চিত্র।



56. TableLayoutPanel: এটা ডিজাইনকালে ব্যবহারের একটি কন্ট্রোল। এটার ওপর আলোচনা ইতোমধ্যে পৃষ্ঠা ১৭৭-এ হয়ে গেছে। সুতরাং এখানে আর কিছু বলার নেই।

57. TextBox: এ কন্ট্রোলটি সম্ভবত সর্বাধিক বেশি ব্যবহৃত হয় ভিজুয়াল বেসিক.নেট প্রোগ্রামিংয়ে। পুরো এ বইটি জুড়ে টেক্সট বাক্স ব্যবহারের উদাহরণ আছে। এখানে অতিরিক্ত যা বলার তাহলো, এটা রিচটেক্সটবক্স থেকে অনেকটা সহজ, এর ভেতর যে কোনো একটিমাত্র ফন্ট ব্যবহার করা যায়, ফন্টের রঙ একই হতে হবে। বিশেষ তেমন কোনো ফর্মাটিংও সম্ভব নয়। মোটকথা বেশি সুবিধার জন্য রিচ টেক্সটবক্স কন্ট্রোল অনেকটা ব্যবহারোপযোগী। টেক্সটবক্স কন্ট্রোলের গুরুত্বপূর্ণ ক'টি প্রোপার্টি হলো: MultiLine, MaxLength, ReadOnly, WordWrap, ScrollBars and Text ।

58. Timer: এটার কাজ হলো কোনো ফাংশন বা প্রসিডুরে ইন্টারভ্যাল বা বিরতি দেওয়া। টাইমার কন্ট্রোল সলুশনের ধূসর এলাকায় পড়ে থাকে। একে কাজে লাগাতে হলে তার বিভিন্ন মেথড ও ফাংশন কল করতে হয় কোডে। একটি উদাহরণ দেখে নিই। নতুন একটি ফর্ম প্রজেক্ট তৈরি করুন। একটি টাইমার কন্ট্রোল প্রজেক্টে নিয়ে আসুন টুলবাক্স থেকে। আরো যুক্ত করুন দু'টি বোতাম ও একটি টেক্সট বাক্স। প্রথম বোতামের টেক্সট ভ্যালু হবে, 'টাইমার শুরু করুন' এবং দ্বিতীয়টির ভ্যালু হবে, 'টাইমার বন্ধ করুন'। টেক্সট বাক্সের ফন্ট সাইজ কিছুটা বাড়িয়ে দিন। এর ভ্যালু প্রপার্টি লিখুন ১। আমার মনে হয় উভয় বোতামের ফন্ট সাইজও কিছুটা বাড়িয়ে দিন যাতে স্পষ্ট দেখা যায়। টাইমারের প্রোপার্টিতে বেয়ে Interval প্রোপার্টির ভ্যালু লিখুন ১০০০। এর অর্থ ১০০০ মিলিসেকেন্ড বা ১ সেকেন্ড। এবার প্রথমে ডবল ক্লিক করুন টাইমার কন্ট্রোলে। কোড এডিটর জানালা আসবে। সেখানে Timer1_Tick ইভেন্ট সাবরুটিন। এ সাবের ভেতর লিখুন:

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    TextBox1.Text = Val(TextBox1.Text) + 1
End Sub
```

১৭তম পরিচ্ছেদ: *Windows Form Controls*

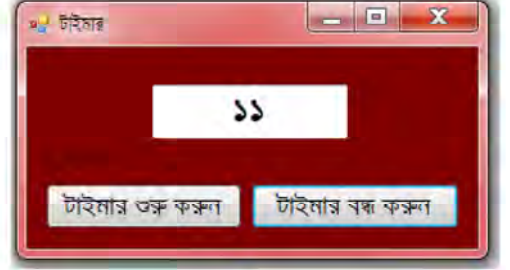
উইন্ডোজ ফর্ম কন্ট্রোলস: *Windows Form Controls*

এরপর 'টাইমার শুরু করুন' বোতামে ডবল ক্লিক করে তার ইভেন্ট সাবে লিখুন:

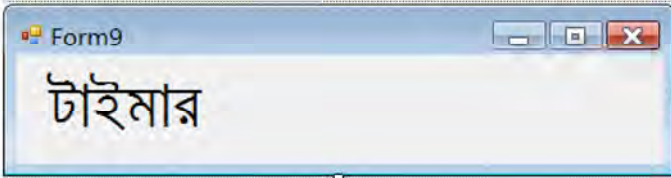
```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    Timer1.Start()
End Sub
```

সবশেষে 'টাইমার বন্ধ করুন' বোতামে ডবল ক্লিক করে তার ইভেন্ট সাবে লিখুন:

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
    Timer1.Stop()
End Sub
```



প্রোগ্রাম শুরু করে উভয় বোতামে ক্লিক করে দেখুন কিভাবে টেক্সট বাক্সে সংখ্যাটি বাড়ে ও বন্ধ হয়। আমরা এই টাইমার কন্ট্রোল দ্বারা একটি ডিজিটাল ঘড়ি বানাতে পারি। দেখুন এটা বানানো কতো সহজ!



নতুন প্রজেক্ট তৈরি করুন। ফর্মে একটি লেবেল নিয়ে আসুন। এর ফন্ট সাইজ ইত্যাদি পরিবর্তন করুন। যেভাবে বায়ের চিত্রে দেখানো হয়েছে। এর ফন্ট হবে বাংলা। আমরা বাংলা ডিজিটাল ঘড়ি বানাবো, কেমন?

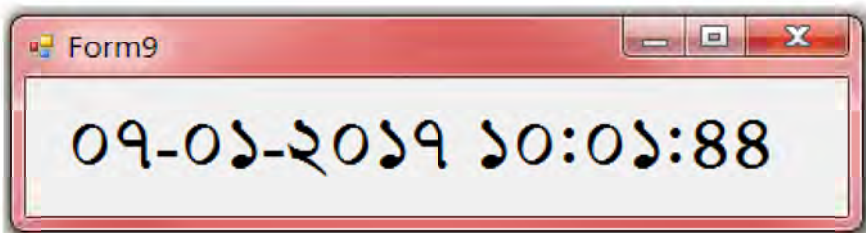
ফর্মের কোথাও ডবল ক্লিক করে এর লোড ইভেন্ট সাবরুটিন প্রদর্শন করুন। এতে লিখুন নিচের কোড লাইনটি। আমরা চাই প্রোগ্রাম শুরু হলেই টাইমার সক্রিয় হয়ে ওঠবে- তাই এই কোড।

```
Private Sub Form9_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Timer1.Enabled = True
End Sub
```

এবার টাইমারে ডবল ক্লিক করে তার ইভেন্ট সাব বের করুন। লিখুন এই কোডটুকু।

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    Label1.Text = Date.Now.ToString("dd-MM-yyyy hh:mm:ss")
    'Other date format "dd MM yyy hh mm ss"
End Sub
```

এ কোডের ওপর কিছুটা ব্যাখ্যার প্রয়োজন। আমরা লেবেলে তারিখ ও সময় দেখাতে চাই। সুতরাং এর টেক্সট-এ এখনকার তারিখ ও সময় [Now] প্রদর্শন করতে নির্দেশ দিচ্ছি। তারিখ ও সময় মূলত অবজেক্ট। ওগুলোকে স্ট্রিং বানাতে যেয়ে লিখতে হয়েছে ToString কীওয়ার্ডটি। ব্রাকেটে আমরা তারিখ ও সময় ফরম্যাট কি হবে সেটা উল্লেখ করেছি। এবার রান করুন। নিচের চিত্রে আমরা ডিজিটাল ঘড়িটি দেখতে পাচ্ছি- আর এটি বাংলা সংখ্যা প্রদর্শন করছে!



১৭তম পরিচ্ছেদ: *Windows Form Controls*

উইন্ডোজ ফর্ম কন্ট্রোলস: *Windows Form Controls*

67. User Control: এতোক্ষণ আমরা ভিজুয়াল বেসিক.নেট এর তৈরি করা কন্ট্রোলগুলোর ওপর বিস্তারিত আলোচনা করেছি। তবে এখানেই শেষ নয়। ব্যবহারকারী বা প্রোগ্রামাররা নিজেরাই নতুন কন্ট্রোল তৈরি করতে পারেন। এগুলোর মাধ্যমে হয়তো বিশেষ কোন কাজও হতে পারে। সুতরাং, আমরা এখন একটি ইউজার কন্ট্রোল তৈরির প্রক্রিয়ার ওপর আলোচনা করবো। বাস্তবে একটি নতুন কন্ট্রোল বানাবো।

মূলত তিন ধরনের ইউজার কন্ট্রোল আছে। যথা:

১. Combination control: এর অর্থ একাধিক ফর্ম কন্ট্রোল দ্বারা একটি নতুন কন্ট্রোল তৈরি করা। সকল উইন্ডোজ কন্ট্রোল এটির ভেতরে থাকবে এবং সবগুলো ফাংশনাল হবে।

২. Inherited control: এটা হলো এমন একটি কন্ট্রোল যা কোনো উইন্ডোজ কন্ট্রোল থেকে উত্তরাধিকার হিসাবে প্রাপ্ত। তবে এতে আরো থাকবে নতুন কিছু ফাংশন।

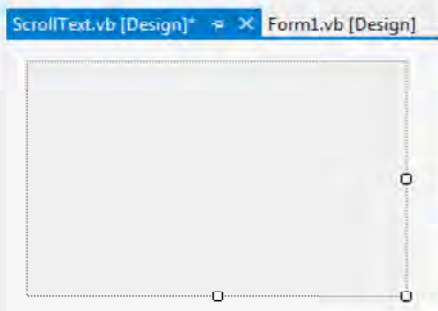
৩. Custom control: এটা সম্পূর্ণ নতুন কন্ট্রোল। এতে কোনো রেডিমেইড কন্ট্রোল থাকবে না।

উপরে বর্ণিত তিনটির মধ্যে প্রথমটির ওপর একটি উদাহরণ এবার তুলে ধরছি। শিক্ষার্থীরা একে একে নিচের পদক্ষেপগুলো অনুসরণ করুন।

১. একটি ফর্ম প্রজেক্ট তৈরি করুন। এর নামকল যাচ্ছেতাই দিতে পারেন।

২. লোড হওয়ার পর চয়েজ করুন: **Project > Add User Control**। এর নামকরণ করুন: **ScrollText**।

৩. এটি আসবে নিচের চিত্রের মতো। টুলবাক্স থেকে একটি লেবেল কন্ট্রোলে নিয়ে আসুন। এর নামকরণ করুন: **LabelScroll**। এরপর যুক্ত করুন একটি টাইমার। টাইমারের ওপর ডবল ক্লিক করে এর **Tick** ইভেন্ট সাব খুলুন। এই সাবের ভেতর লিখুন নিচের কোডগুলো।



```
Public Class ScrollText
    Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
        Dim strBegin As String = Mid(LabelScroll.Text, 1, 1) 'Extract First Character
        Dim strEnd As String = Mid(LabelScroll.Text, 2, LabelScroll.Text.Length()) 'Extract
        Rest Of String
        LabelScroll.Text = strEnd & strBegin 'Add Beginning To End
    End Sub
End Class
```

লক্ষ করুন, এমনিতেই আমাদের কন্ট্রোল নামে একটি ক্লাস এসেছে, ঠিক যে রূপ কোনো ফর্মের ক্লাস সাবরুটিন আসে। এই কোড একটি চলন্ত স্ট্রিং [লেখা] তৈরি করে। লেখার প্রথম অক্ষর ও বাকীগুলো যোগ করে এই কাজটি আঞ্জাম দেয়।

৪. এবার প্রোগ্রামটি বিন্ড করুন: **Menu > Build > Build Solution**। বিন্ড শেষে লক্ষ করুন, টুলবাক্সে আমাদের সৃষ্ট **ScrollText** কন্ট্রোলটিও **User Control Components** লিস্টে এসে গেছে। দেখুন নিচের চিত্রটি।

৫. এবার এই নতুন কন্ট্রোলটি ডবল ক্লিক করে আমাদের উইন্ডোজ ফর্মের ভেতর নিয়ে আসুন।

৬. ফর্মে দু'টি বোতাম যুক্ত করুন। এদের টেক্সট হবে যথাক্রমে: “শুরু” ও “বন্ধ”।

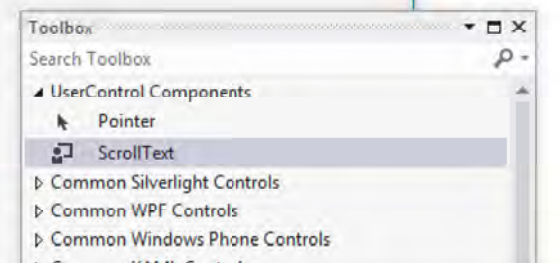
৭. ‘শুরু’ বোতামের ক্লিক সাবরুটিনে এই দু'টি লাইন লিখুন:

```
ScrollText1.LabelScroll.Text = "আমি তো ভাই খুব বেশি বেশি ক্ষুধার্ত..."
Provide Text
ScrollText1.Timer1.Enabled = True 'Enable Timer
```

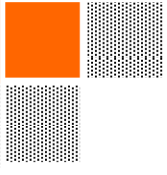
৯. ‘বন্ধ’ বোতামের ক্লিক সাবরুটিনে এই একটি লাইন লিখুন:

```
ScrollText1.Timer1.Enabled = False 'Stop Scrolling
```

ব্যস! আমাদের সৃষ্ট কন্ট্রোল এখন কাজ করবে। রান করে দেখুন।



১৮তম পরিচ্ছেদ: *Visual Basic.Net: Simple Guide*



ভিজুয়াল বেসিক.নেট সহজ গাইড: *Visual Basic.Net: Simple Guide*

এটাই গ্রন্থের সর্বশেষ পরিচ্ছেদ। ভিজুয়াল বেসিক.নেট সম্পর্কে আমার যাকিছু শিখেছি তার ওপর সহজ একটি গাইড এ পরিচ্ছেদে তুলে ধরবো। শিক্ষার্থীরা নিজেদের এ্যাপলিকেশন সৃষ্টিতে এই পরিচ্ছেদকে সময় সময় কাজে লাগাতে পারবেন। এছাড়া মাইক্রোসফটের এই শক্তিশালী প্রোগ্রামিং ভাষার ওপরও এ গাইড থেকে অধিক বেশি জ্ঞানবান হবেন- এটাই আমার বিশ্বাস।

১. সার্বিক ধারণা: **Overview**

Microsoft Visual Basic.Net একটি অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং ভাষা। এর বাস্তবায়ন .নেট [বর্তমান ভার্সন: ৪.৫] নামক ফ্রেমওয়ার্কের ওপর নির্ভরশীল। ভাষার সবকিছুই মূলত বস্তুভিত্তিক। এর মৌলিক টাইপও [Short, Long, Integer, String, Boolean etc.] একেকটি বস্তু। .নেট দ্বারা সৃষ্ট প্রোগ্রাম উইন্ডোজ ছাড়াও লাইনাক্স [Linux] কিংবা ম্যাক অএসএক্স [Mac OSX] অপারেটিং সিস্টেমে রান করা যায়।

২. ডটনেট ফ্রেমওয়ার্ক: **.Net Framework**

এটা মাইক্রোসফটের একটি বৈপ্লবিক সর্বাধুনিক প্রোগ্রামিং ফ্র্যাটফর্ম। এটার দ্বারা প্রোগ্রামাররা সমগ্র বিশ্বব্যাপী প্রতিনিয়ত এ্যাপলিকেশন সৃষ্টি করে যাচ্ছেন:

- (ক) উইন্ডোজ ডেস্কটপ এ্যাপলিকেশন।
- (খ) ওয়েভ এ্যাপলিকেশন ও সার্ভিসেস।

ডটনেট এ্যাপলিকেশন বহুমুখি-ফ্র্যাটফর্মও বটে। এটি ডিজাইন করা হয়েছে বেশ ক'টি সফটওয়্যার ভাষার কথা চিন্তা করে। এগুলোর মধ্যে Visual Basic, C#, C++, Jscript, and COBOL ইত্যাদি এখানে উল্লেখযোগ্য। এ ভাষাগুলো ডটনেট ফ্রেমওয়ার্ক কাজে লাগাতে পারে এবং একে অন্যের সঙ্গে যোগাযোগও সৃষ্টি করতে পারে। ফ্রেমওয়ার্কের মধ্যে একটি বিরাট কোড লাইব্রেরি আছে যা উক্ত ভাষাগুলোর মাধ্যমে ব্যবহার করা যায়। এই গ্রন্থে আলোচিত Vb.Net মূলত লাইব্রেরির কোড ব্যবহারেরই প্রশিক্ষণ মাত্র।

৩. প্রোগ্রাম কাঠামো : **Program Structure**

যে কোন প্রোগ্রামে সাধারণত এই বিষয়গুলো থাকে: ক. নেইমস্পেস ঘোষণা [Namespace declaration] উদাহরণ: Imports System, খ. ক্লাস বা মডিউল [Class or module] উদাহরণ: Public Class Form1; Module Module1, গ. প্রসিডিউর [Procedures- Sub, Function, Operator, Get, Set, AddHandler, RemoveHandler, RaiseEvent], ঘ. ভেরিয়েবল [Variables] উদাহরণ: Dim UserName as String, ঙ. মূল প্রসিডিউর [Main procedure] উদাহরণ: Sub Main(), চ. এক্সপ্রেশন ও স্টেটমেন্ট [Expressions & Statements] উদাহরণ: If MyVar = "Book" then; For x = 1 to 100 এবং ছ. মন্তব্য [Comments] উদাহরণ: 'This is the global variable for keeping User Name। শেষোক্ত [ছ] বিষয়টি বাস্তবে প্রোগ্রামের সঙ্গে সম্পৃক্ত নয়। এটা কম্পাইলার উপেক্ষা করে যাবে। তবে কমেণ্ট থাকলে প্রোগ্রাম বুঝা অনেকটা সহজ হয়। ডিবাগিং ও নতুন সংস্করণ প্রকাশে এগুলো খুব সহায়ক হয়ে থাকে।

৪. ভিজুয়াল বেসিক.নেট মৌলিক সিনট্যাক্স: **VB.Net - Basic Syntax**

আগেই বলেছি ভিজুয়াল বেসিক.নেট একটি পূর্ণাঙ্গ অবজেক্ট ওরিয়েন্টেড ভাষা। এর অর্থ হলো কোনো প্রোগ্রামে থাকবে বিভিন্ন বস্তু বা অবজেক্ট। তারা একে অন্যের সঙ্গে মিথস্ক্রিয়া করবে কিছু ক্রিয়ার মাধ্যমে। আর এসব ক্রিয়ার নামই হচ্ছে পদ্ধতি বা মেথড। একই ধরনের অবজেক্টকে বলে একই টাইপ বা ক্লাসের। সুতরাং আমরা যখন কোনো ভিজুয়াল বেসিক.নেট প্রোগ্রাম নিয়ে আলোচনা করি তখন একে সংজ্ঞায়িত করতে পারি এভাবে: একগুচ্ছ অবজেক্ট বা বস্তু যারা একে অন্যের সঙ্গে যোগাযোগ সৃষ্টি করে যার তার মেথড বা পদ্ধতিসমূহের মাধ্যমে। এসব বস্তু ও মেথডের ওপর আমরা আরো

১৮তম পরিচ্ছেদ: *Visual Basic.Net: Simple Guide*

ভিজুয়াল বেসিক.নেট সহজ গাইড: *Visual Basic.Net: Simple Guide*

কিছু গ্রহণযোগ্য ডিক্লারেশনের উদাহরণ:

```
Dim StudentID As Integer
Dim StudentName As String
Dim Salary As Double
Dim count1, count2 As Integer
Dim status As Boolean
Dim exitButton As New System.Windows.Forms.Button
Dim lastTime, nextTime As Date
```

ভেরিয়েবল ইনিশিয়েলাইজেশন হয় নামের পরে সমান চিহ্ন [=] ও মান লিখে। যেমন: `variable_name = value;`

উদাহরণ:

```
Dim pi As Double
pi = 3.14159
```

প্রয়োজনে ডিক্লারেশনকালেও আমরা ভেরিয়েবল ইনিশিয়েলাইজ করতে পারি। যেমন:

```
Dim StudentID As Integer = 100
Dim StudentName As String = "Asif Talukder"
```

অপরিবর্তনীয় ভেরিয়েবল ডিক্লারেশন: **Declaring Constants** : ভিজুয়াল বেসিক.নেটে যে কোনো স্থানে ভেরিয়েবলের মতো কনস্ট্যান্ট ভেরিয়েবল ডিক্লারার করা যায়। এগুলো ঠিক ভেরিয়েবলের মতোই, কিন্তু এদের মান পরিবর্তন হয় না। একটি উদাহরণ দেখি।

```
Module constantsNenum
    Sub Main()
        Const PI = 3.14149 'here is the constant declaration
        Dim radius, area As Single
        radius = 7
        area = PI * radius * radius
        Console.WriteLine("Area = " & Str(area))
        Console.ReadKey()
    End Sub
End Module
```

উপরের প্রোগ্রাম রানের ফলাফল: Area = 153.933।

১০. প্রিন্ট ও প্রদর্শন কনস্ট্যান্ট: **Print and Display Constants**

ভিজুয়াল বেসিক.নেট আমাদেরকে নিচের কনস্ট্যান্টগুলো তৈরি করে দিয়েছে।

Constant	Description
vbCrLf	Carriage return/linefeed character combination.
vbCr	Carriage return character.
vbLf	Linefeed character.
vbNewLine	Newline character.
vbNullChar	Null character.
vbNullString	Not the same as a zero-length string (""); used for calling external procedures.
vbObjectError	Error number. User-defined error numbers should be greater than this value. For example: <code>Err.Raise(Number) = vbObjectError + 1000</code>
vbTab	Tab character.
vbBack	Backspace character.

১৮তম পরিচ্ছেদ: *Visual Basic.Net: Simple Guide*

ভিজুয়াল বেসিক.নেট সহজ গাইড: *Visual Basic.Net: Simple Guide*

S.N	Statements and Description	Example
3	Enum Statement Declares an enumeration and defines the values of its members.	Enum CoffeeMugSize Jumbo ExtraLarge Large Medium Small End Enum
4	Class Statement Declares the name of a class and introduces the definition of the variables, properties, events, and procedures that the class comprises.	Class Box Public length As Double Public breadth As Double Public height As Double End Class
5	Structure Statement Declares the name of a structure and introduces the definition of the variables, properties, events, and procedures that the structure comprises.	Structure Box Public length As Double Public breadth As Double Public height As Double End Structure
6	Module Statement Declares the name of a module and introduces the definition of the variables, properties, events, and procedures that the module comprises.	Public Module myModule Sub Main() Dim user As String = InputBox("What is your name?") MsgBox("User name is" & user) End Sub End Module
7	Interface Statement Declares the name of an interface and introduces the definitions of the members that the interface comprises.	Public Interface MyInterface Sub doSomething() End Interface
8	Function Statement Declares the name, parameters, and code that define a Function procedure.	Function myFunction (ByVal n As Integer) As Double Return 5.87 * n End Function
9	Sub Statement Declares the name, parameters, and code that define a Sub procedure.	Sub mySub(ByVal s As String) Return End Sub
10	Declare Statement Declares a reference to a procedure implemented in an external file.	Declare Function getUsername Lib "advapi32.dll" Alias "GetUserNameA" (ByVal lpBuffer As String, ByRef nSize As Integer) As Integer
11	Operator Statement Declares the operator symbol, operands, and code that define an operator procedure on a class or structure.	Public Shared Operator + (ByVal x As obj, ByVal y As obj) As obj Dim r As New obj ' implementation code for r = x + y Return r End Operator

