RESEARCH ARTICLE OPEN ACCESS

Visual Studio Professional 2012: A Benchmark for Tomorrow

Devadrita Dey Sarkar
Amity University
Sector 125, Noida - 20130
Uttar Pradesh – India

ABSTRACT
This paper aims at the basic idea behind visual studio 4.0(2012 version). The definitive guide to using Visual C# .NET to develop stand-alone applications for Microsoft Windows and Web-enabled Microsoft .NET applications Get the complete guidance we need to use the Visual C# .NET language to produce Windows-based applications and Web-enabled .NET applications with this comprehensive reference. It thoroughly covers the language's structure, syntax, code wizards, and the Microsoft Visual Studio design environment, paying close attention to both the client and server sides of the .NET environment. We'll find detailed answers and best practices to help one write, test, and debug applications and extend them to the Web-quickly and intuitively. An extensive collection of real-world programming examples demonstrates solutions to specific coding problems.

Keywords:- C#,.Net, Microsoft, Syntax, Client-Server.

I. INTRODUCTION

The Microsoft **.NET Compact Framework** (.NET CF) is a version of the .NET Framework that is designed to run on resource constrained mobile/embedded devices such as personal digital assistants (PDAs), mobile phones, factory controllers, set-top boxes, etc.[1] The .NET Compact Framework uses some of the same class libraries as the full .NET Framework and also a few libraries designed specifically for mobile devices such as .NET Compact Framework controls. However, the libraries are not exact copies of the .NET Framework; they are scaled down to use less space.[2]

It is possible to develop applications that use the .NET Compact Framework in Visual Studio .NET 2003, in Visual Studio 2005 and in Visual Studio 2008, in C# or Visual Basic .NET. Applications developed with Basic4ppc are also eventually compiled for the .NET CF. The resulting applications are designed to run on a special, mobile-device, high performance JIT compiler.

The UI development is based on Windows Forms which is also available on the desktop version of the .NET Framework.

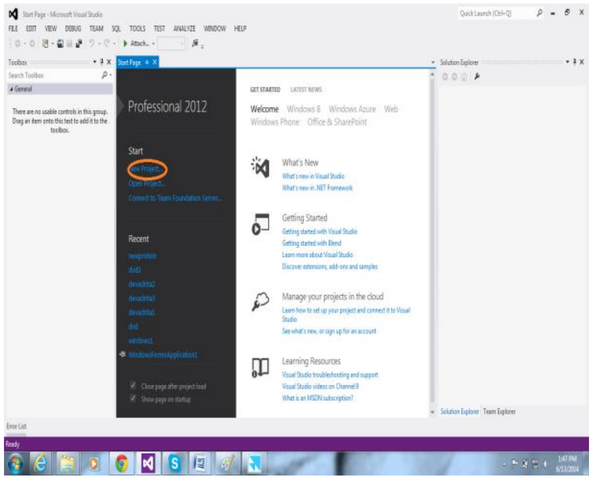

There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.
Start Page - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
FILE EDIT VIEW DEBUG TEAM SQL TOOLS TEST ANALYZE WINDOW HELP
Attach...
Toolbox
Search Toolbox
General
Start Page
Solution Explorer
Professional 2012
Start
New Project...
Open Project...
Connect to Team Foundation Server...
Recent
GET STARTED
LATEST NEWS
Welcome Windows 8 Windows Azure Web
Windows Phone Office & SharePoint
What's New
What's new in Visual Studio
What's new in .NET Framework
Getting Started
Getting started with Visual Studio
Getting started with Blend
Learn more about Visual Studio
Discover extensions, add-ons and samples
Manage your projects in the cloud
Learn how to set up your project and connect it to Visual Studio
See what's new, or sign up for an account
Learning Resources
Visual Studio troubleshooting and support
Visual Studio videos on Channel 9
What is an MSDN subscription?
Close page after project load
Show page on startup
Solution Explorer Team Explorer
Error List
Ready

Figure 1:homepage of visual studio4.0

II. RELATED WORK

User interfaces can easily be created with Visual Studio by placing .NET Compact Framework controls like buttons, text boxes, etc. on the forms. Also features like data binding are available for the .NET CF. A major disadvantage of the UI development is that modern looking applications with support for finger-based touch screen interaction are not that easy to implement. This is mainly due to the desktop-oriented user interface concept on which Windows Forms is based, although some third party libraries with custom controls for this purpose are available.[3]

III. IMPLIMENTATION

C# is a multi-paradigm programming language encompassing strong typing , imperative,declarative, functional, generic, obje ct-oriented (class-based), and component-oriented programming disciplines.[4] It was developed by Microsoft within its .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270:2006). C# is one of the programming languages designed for the Common Language Infrastructure. C# is built on the syntax and semantics of C++, allowing C programmers to take advantage of .NET and the common language runtime.

C# has the following syntax:

- Semicolons are used to denote the end of a statement.
- Curly braces are used to group statements. Statements are commonly grouped into methods (functions), methods into classes, and classes into namespaces.[5]
- Variables are assigned using an equals sign, but compared using two consecutive equals signs.

DISTINGUISHING FEATURES OF C#:

- C# supports strongly typed implicit variable declarations with the keyword **var**, and implicitly typed arrays with the keyword new followed by a collection initializer.
- Meta programming via C# attributes is part of the language. Many of these attributes duplicate the functionality of GCC's and VisualC++'s platform-dependent preprocessor directives.[6]
- Like C++, and unlike Java, C# programmers must use the keyword **virtual** to allow methods to be overridden by subclasses.
- *Extension methods* in C# allow programmers to use static methods as if they were methods from a class's method table, allowing programmers to add methods to an object that they feel should exist on that object and its derivatives.[8]
- The type **dynamic** allows for run-time method binding, allowing for JavaScript like method calls and run-time object composition.[7]
- C# has support for strongly-typed function pointers via the keyword **delegate**.
- Like the Qt framework's pseudo-C++ *signal* and *slot*, C# has semantics specifically surrounding publish-subscribe style events, though C# uses delegates to do so.[9]

C# offers Java-like **synchronized** method calls, via the attribute [MethodImpl(MethodImplOptions.Synchronized), and has support for mutually-exclusive locks via the keyword **lock**.

IV. SAMPLE WEB APPLICATION OF VISUAL STUDIO 4.0

TASK 1: To design a window application containing two labels, one textbox and a button.label2 must be hidden but its textbox can have a tag-line of our choice and the display of the button must be “hello!”.

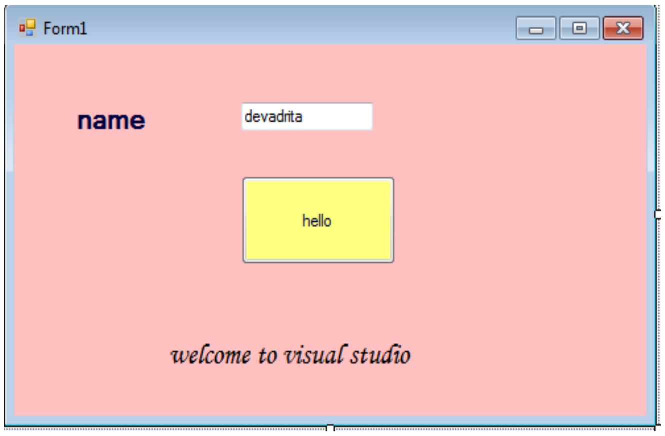
# CODING:

```
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace windows1

 public partial class Form1 : Form
➢   {
➢  public Form1()
➢      {
➢    InitializeComponent();
➢      }
➢  private void Form1_Load(object sender,
 EventArgs e)
➢      }
➢ private void button1_Click(object sender,
 EventArgs e)
➢      {
```

```
➢            label2.Visible= true;
➢       }
➢           private void textBox2_TextChanged(object
     sender, EventArgs e)
➢        }
➢          label1.Visible = false;
      }
```

OUTPUT FOR TASK1:



TASK 2: To design a window form application to display curriculum vitae of a person by opening a panel and installing three labels and three textbox and two buttons for saving and opening.

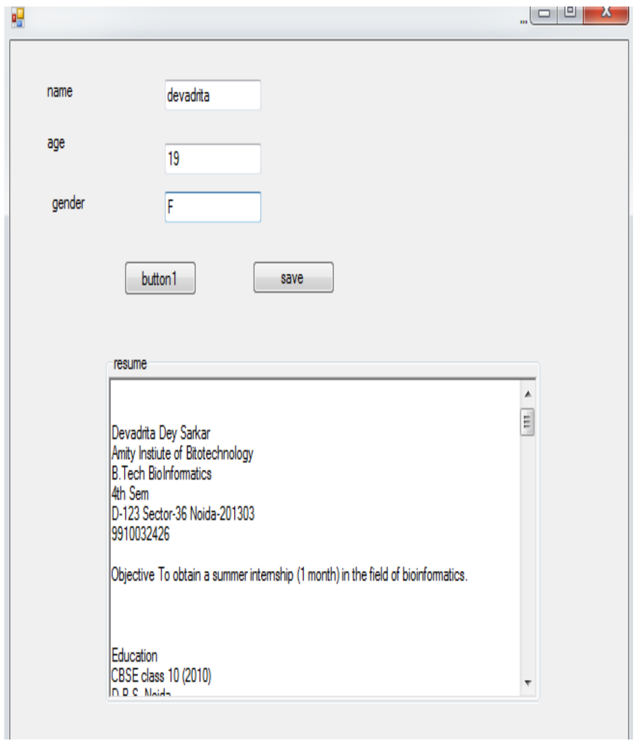
```
➢ #CODING:
➢  using System;
➢ using System.Collections.Generic;
➢ using System.ComponentModel;
➢ using System.Data;
➢ using System.Drawing;
➢ using System.Linq;
➢ using System.Text;
➢ using System.Threading.Tasks;
➢ using System.Windows.Forms;
➢ using System.IO;
➢ namespace dvd3
➢ {
➢     public partial class Form1 : Form
➢     {
➢     public Form1()
➢       {
➢        InitializeComponent();
➢         }
➢      private void button1_Click(object sender,
   EventArgs e)
➢        {
➢            DialogResult rs = op.ShowDialog();
➢            if (rs == DialogResult.OK)
➢            {
➢               string s = op.FileName;
➢               StreamReader sr = new
   StreamReader(s);
➢               richTextBox1.Text = sr.ReadToEnd();

➢               sr.Close();

➢   }
➢        }
➢
➢        private void
   richTextBox1_TextChanged(object sender,
   EventArgs e)
➢        {
➢
➢        }
➢
➢        private void save_Click(object sender,
   EventArgs e)
➢        {
➢           if (sv.ShowDialog() == DialogResult.OK)
➢           {
➢               using (Stream s =
   File.Open(sv.FileName, FileMode.CreateNew))
➢               {
➢                  StreamWriter sw = new
   StreamWriter(s);
➢                  sw.WriteLine(richTextBox1.Text);
➢                  sw.Close();
➢               }
➢               string OnlyFileName=
   sv.FileName.Substring(sv)
➢           }

➢        }
```

OUTPUT FOR TASK2:



name devadrita
age 19
gender F

resume

Devadrita Dey Sarkar
Amity Instiute of Bitotechnology
B.Tech BioInformatics
4th Sem
D-123 Sector-36 Noida-201303
9910032426

Objective To obtain a summer internship (1 month) in the field of bioinformatics.

Education
CBSE class 10 (2010)

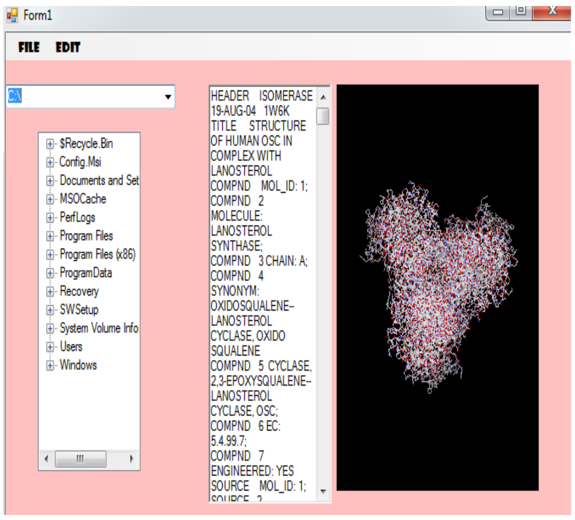
TASK3: To design a windows form application to open a protein pdb file and its structure with the help of RASMOL active x control, a panel, a tree view, a rich text box and a ras control key. Menu strip, save dialogbox and open file dialog box are also generated and to display the list of drives in the tree view through combo box.

```
➢ # CODING:
➢  using System;
➢ using System.Collections.Generic;
➢ using System.ComponentModel;
➢ using System.Data;
➢ using System.Drawing;
➢ using System.Linq;
➢ using System.Text;
➢ using System.Threading.Tasks;
➢ using System.Windows.Forms;
➢ using System.IO;
➢
➢ namespace newprotein
➢ {
➢     public partial class Form1 : Form
➢     {
➢         public Form1()
➢         {
➢             InitializeComponent();
➢             foreach (var Drives in
   Environment.GetLogicalDrives())
➢             {
➢                 DriveInfo DriveInf = new
   DriveInfo(Drives);
➢                 if (DriveInf.IsReady == true)
➢                 {
➢
   comboBox1.Items.Add(DriveInf.Name);
➢                 }

➢             }
➢             private void op_FileOk(object sender,
   CancelEventArgs e)
➢         {
➢
➢         }
➢
➢         private                              void
   openToolStripMenuItem_Click(object     sender,
   EventArgs e)
➢         {
➢             DialogResult result = op.ShowDialog();
➢             string path = op.FileName;
➢             if (result == DialogResult.OK)
➢             {
➢                 StreamReader       sr       =       new
   StreamReader(path);
➢                 richTextBox1.Text = sr.ReadToEnd();
➢                 sr.Close();
➢                 axRasCtrl1.ExecuteCommand(1,  "Load
   " + path);
➢
➢             }
➢         }
➢
➢         private void
   saveToolStripMenuItem_Click(object sender,
   EventArgs e)
➢         {
➢             if (sd.ShowDialog() == DialogResult.OK)
➢             {
➢                 using (Stream s =
   File.Open(sd.FileName, FileMode.Create))
➢                 {
➢                     StreamWriter sw = new
   StreamWriter(s);
➢                     sw.WriteLine(richTextBox1.Text);
➢                     sw.Close();
➢
➢                 }
➢             }
➢         }
➢
```

```
➢      private void
exitToolStripMenuItem_Click(object sender,
EventArgs e)
➢      {
➢         this.Close();
➢         //class newprotein.Form1
➢       }
➢
➢      private void
comboBox1_SelectedIndexChanged(object
sender, EventArgs e)
➢      {
➢         treeView1.Nodes.Clear();
➢
➢         DirectoryInfo path =new
DirectoryInfo(comboBox1.SelectedItem.ToStrin
g());
➢
➢         try
➢         {
➢            foreach (DirectoryInfo dir in
path.GetDirectories())
➢            {
➢               TreeNode node =
treeView1.Nodes.Add(dir.Name);
➢               node.Nodes.Add(" ");
➢            }
➢            foreach(FileInfo file in path.GetFiles())
➢            {
➢               if (file.Extension.ToLower() ==
".txt")
➢               {
➢                  TreeNode node =
treeView1.Nodes.Add(file.Name);
➢               }
➢            }
➢      }
➢         catch (Exception ex)
➢         {
➢            MessageBox.Show(ex.Message);
➢         }
➢      }
```

OUTPUT FOR TASK3:



```
$Recycle.Bin
Config.Msi
Documents and Set
MSOCache
PerfLogs
Program Files
Program Files (x86)
ProgramData
Recovery
SWSetup
System Volume Info
Users
Windows
```

```
HEADER   ISOMERASE
19-AUG-04   1W6K
TITLE    STRUCTURE
OF HUMAN OSC IN
COMPLEX WITH
LANOSTEROL
COMPND   MOL_ID: 1;
COMPND  2
MOLECULE:
LANOSTEROL
SYNTHASE;
COMPND  3 CHAIN: A;
COMPND  4
SYNONYM:
OXIDOSQUALENE--
LANOSTEROL
CYCLASE, OXIDO
SQUALENE
COMPND  5 CYCLASE,
2,3-EPOXYSQUALENE--
LANOSTEROL
CYCLASE, OSC;
COMPND  6 EC:
5.4.99.7;
COMPND  7
ENGINEERED: YES
SOURCE   MOL_ID: 1;
```

V. FUTURE WORK

As far as the future aspects are concerned Microsoft is innovating for the future of .NET. Announcements at Build 2014, Microsoft's developer conference, previewed updates at the core of .NET – the basis of business applications for the web, Windows desktop, scalable cloud services through Microsoft Azure, or for reaching end users via Windows Store devices or other cross-device development strategies.[10]

REFERENCES

[1] "Visual Studio 2012 Update 4 (2013.4) RTM". Retrieved 2014-11-12.

[2] Lextrait, Vincent (January 2010). "The Programming Languages Beacon, v10.0". Retrieved 5 January 2010.

[3] Guthrie, Scott. "Releasing the Source Code for the .NET Framework Libraries". Retrieved 2007-10-04.

[4] Brenner, Pat (19 July 2013). "C99 library support in Visual Studio 2013". *Visual C++* Team Blog. Microsoft. Retrieved 3 August 2014.

[5] Microsoft Releases Visual Studio 2010, .NET Framework 4.

[6] Microsoft Releases Visual Studio 2012, .NET Framework 4

[7] Visual Studio Ultimate with MSDN". *visualstudio.com*. Microsoft. System Requirements. Retrieved 10 November 2014.

[8] Quintero, Carlos. "Visual Studio 2010 Extensibility moving beyond add-ins and packages". Retrieved 2009-04-18.

[9] Visual Studio 2012 and .NET 4.5 Complete! - Somasegar's blog - Site Home - MSDN Blogs". Blogs.msdn.com. 2012-08-01. Retrieved 2013-06-15.

[10]. *Visual Studio portal*. Microsoft. Retrieved30 November 2012.