**Unit 1 Introduction**

**A brief Description of Visual Basic 2010**

Visual Basic 2010 is the latest version of Visual Basic launched by Microsoft in 2010. It is almost similar to Visual Basic 2008 but it has added many new features. Visual Basic has gone through many phases of development since the days of BASIC that was built for DOS. BASIC stands for Beginners' All-purpose Symbolic Instruction Code. The program code in Visual Basic resembles the English language. Different software companies had produced many different versions of BASIC for DOS, such as Microsoft QBASIC, QUICKBASIC, GWBASIC, and IBM BASICA and more. Then, Microsoft launched the first graphical BASIC, Visual Basic Version 1 in 1991. It is GUI based and especially developed for MS window. Since then Microsoft slowly phased out the DOS versions of BASIC and completely replaced them by Visual Basic.

Visual Basic was initially a functional or procedural programming language until the

popular Visual Basic 6. Then, Microsoft transformed Visual Basic into a more powerful object oriented programming language by launching Visual Basic.Net, Visual Basic 2005, Visual Basic 2008 and the latest Visual Basic 2010. Visual Basic 2010 is a full-fledged Object-Oriented Programming (OOP) Language; it has caught up with other OOP languages such as C++, Java, C# and others.

**Introduction of .NET**

Sometime in the July 2000, Microsoft announced a whole new software development framework for Windows called .NET in the Professional Developer Conference (PDC). Microsoft also released PDC version of the software for the developers to test. After initial testing and feedback Beta 1 of .NET was announced. Beta 1 of the .NET itself got lot of attention from the developer community. When Microsoft announced Beta 2, it incorporated many changes suggested by the community and internals into the software. The overall 'Beta' phase lasted for more than 1 ½ years. Finally, in March 2002 Microsoft released final version of the .NET framework.

When you hear the name .NET, it gives a feeling that it is something to do only with internet or networked applications. Even though it is true that .NET provides solid foundation for developing such applications it is possible to create many other types of applications

The first version .NET is released with the name .NET framework n1.0 on February 2002.alog with visual studio 2002 IDE. The second version of .NET is released after one year and was known as .NET framework 1.1. Microsoft visual studio .NET better is known as .NET 2003, was also a part of the second release .NET framework 2.0 was released with visual studio 2005 on November 07 2005. after the .NET 3.0 formerly called WinFX. was then released on November 06, 2006, .NET framework 4.0, was announced by Microsoft

on September 29, 2008 and on April 12, 2010, the final version of .NET Framework 4.0 was released.

**Features of .NET**

Now that we know some basics of .NET, let us see what makes .NET a wonderful

platform for developing modern applications.

**• Rich Functionality out of the box**

.NET framework provides a rich set of functionality out of the box. It contains hundreds of classes that provide variety of functionality ready to use in your applications. This means that as a developer you need not go into low level details of many operations such as file IO, network communication and so on.

• **Easy development of web applications**

ASP.NET is a technology available on .NET platform for developing dynamic and data driven web applications. ASP.NET provides an event driven programming model (similar to Visual Basic 6 that simplify development of web pages (now called as web forms) with complex user interface. ASP.NET server controls provide advanced user interface elements (like calendar and grids) that save lot of coding from programmer’s side.

• **OOPs Support**

The advantages of Object Oriented programming are well known. .NET provides a fully object oriented environment. The philosophy of .NET is – “Object is mother of all.” Languages like Visual Basic.NET now support many of the OO features that were lacking traditionally. Even primitive types like integer and characters can be treated as objects – something not available even in OO languages like C++.

**• Multi-Language Support**

Generally enterprises have varying skill sets. For example, a company might have people with skills in Visual Basic, C++, and Java etc. It is an experience that whenever a new language or environment is invented existing skills are outdated. This naturally increases cost of training and learning curve. .NET provides something attractive in this area. It supports multiple languages. This means that if you have skills in C++, you need not throw them but just mould them to suit .NET environment. Currently four languages are available right out of the box namely – Visual Basic.NET, C# (pronounced as C-sharp), Jscript.NET and Managed C++ (a dialect of Visual C++). There are many vendors that are working on developing language compilers for other languages (20+ language compilers are already available). The beauty of multi language support lies in the fact that even though the syntax of each language is different, the basic capabilities of each language remain at par with one another.

• **Multi-Device Support**

Modern lift style is increasingly embracing mobile and wireless devices such as PDAs, mobiles and handheld PCs. . .NET provides promising platform for programming such devices. .NET Compact Framework and Mobile Internet Toolkit are step ahead in this direction.

• **Automatic memory management**

While developing applications developers had to develop an eye on system resources like memory. Memory leaks were major reason in failure of applications. .NET takes this worry away from developer by handling memory on its own. The garbage collector takes care of freeing unused objects at appropriate intervals.

• **No more DLL Hell**

If you have worked with COM components, you probably are aware of “DLL hell”. DLL conflicts are a common fact in COM world. The main reason behind this was the philosophy of COM – “one version of component across machine”. Also, COM components require registration in the system registry. .NET ends this DLL hell by allowing applications to use their own copy of dependent DLLs. Also, .NET components do not require any kind of registration in system registry.

• **Strong XML support**

Now days it is hard to find a programmer who is unaware of XML. XML has gained such a strong industry support that almost all the vendors have released some kind of upgrades or patches to their existing software to make it “XML compatible”. Currently, .NET is the only platform that has built with XML right into the core framework. .NET tries to harness power of XML in every possible way. In addition to providing support for manipulating and transforming XML documents, .NET provides XML web services that are based on standards like HTTP, XML.

**• Security**

Windows platform was always criticized for poor security mechanisms. Microsoft has taken great efforts to make .NET platform safe and secure for enterprise applications. Features such as type safety, code access security and role based authentication make overall application more robust and secure.

**The *.NET Platform Architecture***

The .NET is a framework that covers all the layers of software development above the Operating System. It provides the richest level of integration among presentation technologies, component technologies, and data technologies ever seen on Microsoft, or perhaps any, platform. Secondly, the entire architecture has been created to make it easy to develop Internet applications, as it is to develop for the desktop.

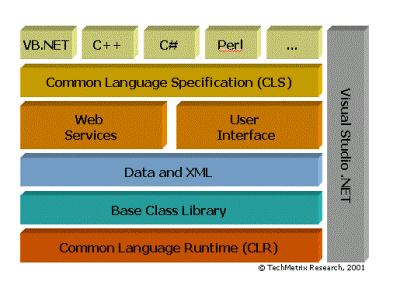
**Constituents of .NET Platform**

The .NET consists of the following three main parts

.NET Framework – a completely re-engineered development environment.

.NET Products – applications from MS based on the .NET platform, including Office and Visual Studio.

.NET Services – facilitates 3rd party developers to create services on the .NET Platform.



Operating System

New version of .NET provides a Framework

that helps in designing portable, scalable and robust applications. The application developed in .NET framework 4.0 can be executed in a distributed environment. Therefore you can say that the .NET framework 4.0 can be designed to address the latest needs of the development.

**Common Language Runtime:**

One of the most important component of .NET framework is the CLR, better known as the runtime. .NET applications are compiled to a common language known as Microsoft Intermediate Language or "IL". The CLR, then, handles the compiling the IL to machine language, at which point the program is executed.

It provides functionalities such as memory management, exception handling, debugging, security, thread execution, code execution, code safety, verification, and compilation. It also support versioning to any language that the CLR interacts with. This implies that CLR can host a variety of language and provide common tools to these language, there, by ensuring interoperability between the codes.

The runtime manages the execution of the code. Therefore the code that works is called the managed code. CLR also supports services that the application uses to access various resources such as collection, arrays, and operating system folders. The runtime automatically releases the objects when they are no longer in use.

The runtime imposes Code Access Security(CAS), which is the CLR ’s security system that enforces security policies by preventing unauthorized access to protected and operations. CAS also helps in making the code robust, which implies that it makes the code extensible and reusable. It implements a strict and code –verification infrastructure called CTS.

The runtime also accelerates the developer’s productivity as the developers can write the application in any language such as c#, visual basic, visual c++.

**Managed code :**  managed code is the code that is executed directly by the CLR. The applications that are created using managed code automatically have CLR services, such as type checking, security, and automatic garbage collection. These services help to provide platform and language independence to managed code applications. The CLR compiles the application to intermediate language(IL) and not the machine code. This IL along with the metadata describes the attributes, classes and methods of the code reside in an assembly. The compilation takes place in the managed execution environment, which assures the working code.

A piece of managed code executed as follows

a. selecting a language

b. compiling the code to IL

c. compiling IL to native code

d. executing the code

**Unmanaged Code :-** in contrast the managed code, unmanaged code directly complies to the machine code and runs on the machine where it has been compiled. It does not have services such as security or memory management.

**Memory management :** one of the most important services that the CLR provides during managed execution is the automatic memory management. The CLR uses the garbage collector to manage the allocation and release of memory for an application.

**.NET Framework class Library :**

The next layer up in the framework is called the .NET Class Framework also referred as .NET base class library. The .NET Class Framework consists of several thousand type definitions, where each type exposes some functionality.

It facilitate interoperability between language, . NET framework should be CLS (common language specification) compliant so that they can used from any programming language whose compiler conforms to the CLS.

NET application, components, and controls are built on the foundation of .NET framework types. These entities perform the following functions.

1. Represent base data types and exceptions
2. Encapsulating data structure.
3. Perform input and output operation etc

The framework class library (FCL) is a huge library of reusable types meant to use by managed code. It is an object-oriented library that is used in component based application. The FCL is made up of a hierarchy of namespaces that expose classes, structures, interfaces, enumerations. Names spaces are logically defined by their functionality. For example the **System.Data** namespace contains the functionality available for accessing database. There are more than 20,000 classes in FCL. All logically grouped in a hierarchical manner .

The following points need to be remembered while using the FCL.

1. The classes, interfaces, structures and enumerated values collectively referred to as type.
2. The different types in the framework are arranged in a hierarchy of namespaces. This solves the problem faced in name collisions.
3. We may write a code that uses a ***stream*** class in IL, these types are represented as **System.IO.Stream.**

**Metadata and Assemblies :** metadata is library information that describes your program stored in a CLR portable executable(PE) file or the memory. When compilation of the code takes place in a PE file, the metadata is inserted into one-part of the file, while the code is converted into IL and inserted into the other part of the file. The metadata describes every type and member. When the code is in the run mode, the CLR loads the metadata into the memory and finds information about the code’s class and members. A metadata describes information about the code in a language-neutral manner. The metadata contains the following.

1. Assembly information, which includes the metadata identity that can be name, version, culture, public key, other referenced assemblies, and security permissions.
2. Information about type such as name, visibility, base classes ,interfaces used members.
3. Attribute information that modifies the types and members of a class.

After providing the listed information, CLR is able to create objects, access data and call members.

Assemblies are packaged into units containing programs and libraries. An assebly contains a self-describing binary file that can be either Dynamic link library file or executable file. An asseble contains code that CLR executes.

Asseblies can be of two types, static or dynamic. Static assemblies include interfaces, classes and resources. These asseblies are saved in the PE on a disk. Other hand dynamic asseblies run directly from the memory without being saved to disk before execution.

**Assembly manifest :** every assembly contains data that describes how elements are related to each other withing assembly. the manifiest also known as assebly manifest, contains the assembly needed for providing the assembly's verson requirements and security identity. the menifiest can be stored either in a PE with IL code or in a standalone PE file that contains only the manifiest information table.

|  |  |
| --- | --- |
| assebly name | name of the assembly |
| version number | -indicate the majore version number, a minor version number |
| culture | provide the language name that assembly support. |
| list of file in assembly | it contains the list of file names that contains in assebly name. |
| type reference information | contains the information required to map a type reference to the file that contains its declaration and implementation. |

**Global Assebly Catche: GAC**  is the central place for registering assemblies, so that different application on the computer can use it later on. CLR is installed on the achine-wide codecache in a computer.

**Strong Name Asseblies:**  A strong name assemblies the assembly's identity, that is the information about the assembly's name, version number, culture, public key, and digital signature.

**private and shared asseblies:** when a single computer uses an assembly then it is called as a private assembly.

shared assemblies are those assemblies that are placed in the global assembly cache so that they can be used by multiple applications.

**Side by side exxecution :**  the process of executing multiple versions of an application or an assembly is known as side-by-side execution.

**LINQ** :- LINQ is one of the components of .NET framework 4.0. that adds native data query capabilities to .NET language by using systax similar to SQL.

**The** following is a simple query in VB.

**Dim query = from s in students where s.gender=”m”**

**Select s.name**

When you execute the preceding query, it returns thename of all the male students and is stored in the variable query. To print that list, you have to write the following lines of code.

**For each name is query**

**Response.write(name)**

**Next**

**It is import to note that it is looks similar to SQL. The LINQ query displayed is not an SQL statement. Instread it is purely VB language.**

**Though LINQ queries resembles SQL, they are not restricted to access only relational database. LINQ enabled data access componets are as follows.**

1. **LINQ to ADO.NET** : includes two options to SQL which translates a query into an SQL query and then issues it against the tables specified in the query in an SQL server database and LINQ to Data Set which execute the query on the components of a **DataSet**.
2. **LINQ to Object :- allows** querying objects in a collection. LINQ to Objects is not dynamic. After creating and using the result set, any changes made to the source collection do not automatically update the result set.
3. **LINQ to SQL:** allows querying of XML data. In addition, it helps in creating and manipulating XML data. This option has a different syntax. The basic LINQ remains the same.

**Exploring New Features of ..NET Framework 4.0**

**.NET framework** 4.0 helps in providing a consistent object oriented programming environment. It helps the developers to exerience consistent programming environment. There are many new features in this frame work.

1. **Improved application compatibility and deployment support:** The application that are developed in the earlier versions of .net framework are supported in this version. This implies that the new version of .NET framework in compatible with the applications that are developed using the earlier version of .net framework.

**.NET Framework client** **profile :** improves the deployment process by providing a fast deployment of windows, forms, console, and in process side by side execution.

**In-Process Side-by-**Side Execution :- helps in loading and starting multiple versioning of .NET Framework. This implies you can run multiple of CLR in a single process.

1. **Dynamic language Runtime (DLR):-** it is use to implement dynamic languages, such as Python and Ruby, on the .NET Framework. The dynamic languages are the commonly used scripting languages and the users can extend the applications, which are created using dynamic language.
2. **Managed extensibility Framework:-** the MEF is a new library included in framework 4.0 to create lightweight and extensible applications. In other words it helps in simplifying the creations of extensible application.
3. **Parallel Computing:-** this features allowed to run multiple thread simultaneously. It means, large program can be divided into small-2 program and all these programs can run at same time.
4. **Improved Security:-** there are many changes made on code access security(CAS) system. There are many useful classes and services introduced in CLR and .NET framework that help the developers to write secure code and implement cryptography and role based security**.**
5. **Networking Improvements :-**  there are various improvements made on networking .
6. **Windows authentication like as System.Net.HttpWebRequest. System.Net.HttpListener, etc**
7. **It supports Ipv6 internet protocol.**
8. **Introduction of Add Range in the System.Net.HttpWebRequest.class.**
9. **Improved Entity Framework** :- in this version of .NET framework. There is less amount of coding in entity framework. The amount maintenance required for data driven application has also been reduced by allowing we to program against data models. It also support to create N Tier application.

**Visual Studio 2010 IDE**

Microsoft Visual Studio is an Integrated Development Environment (IDE) provided by Microsoft to create and develop Window-based, web-based, console-based and mobile-based application in .NET Framework. These application might be created by using different languages, such as visual Basic, C#, and Visual C++. You can also use Notepad as a text editor to create these applications. For this you need a compiler that could convert the programming language into machine language.

Visual Studio 2010 is the latest version of Visual Studio that targets .NET Framework 4.0. It helps to minimize the development time of .NET applications by providing different tools to integrate designers into the overall development process.

The following are some of the benefits of using Visual Studio 2010:

* Helps in minimizing the application development time.
* Simplifies the process of testing applications.
* provides different toolsets for integrating graphics designers into the overall development process.
* Supports multiple versions of the .NET Framework.
* Enhances data retrieval and data binding.
* Helps in finding sections of code that need refactoring.
* Provides support for Web, Mobile, Windows, and Office application development.
* Integrates with the .NET Framework 4.0.

**Developing Visual Studio 2010 Application (Project)**

In Visual Studio 2010, you can develop mainly the following types of applications:

* Console applications
* Window Forms application
* Web applications

**Developing a Console Application**

A console application refers to a .NET application that does not have Graphical User Interface (GUI) except for a command-line console. The users of a console application interact with the application only through the command line console, implying the data is displayed and read through the command line. This makes console application suitable for situations where less user interaction is required. Visual Studio provides a project template which offers you an easy way to develop console application.

Eg:

Module Module1

Sub Main()

Console.WriteLine("Wel Come to Visual Studio 2010")

Console.ReadLine()

End Sub

End Module

The **WriteLine** method of the **Console** class is used to display a message on the command-line console. When this line of code is executed, the message, First Console Application, is displayed on the command-line console.

The **ReadLine** method of the **Console class** is used to read the next line of characters from the console.

Imports System.Console

Module Module1

Sub Main()

WriteLine("Welcome to Visual Studio 2010")

Read()

End Sub

End Module

**Developing a Windows Forms Application**

Windows Forms applications refer to those applications that have a form at the center of the application. A form in a Windows Forms application is a rectangular area that has a title bar at the top and minimize, maximize and close buttons at the upper-right corner. It provides GUI for the application thereby allowing interaction with the user.

eg:

Public Class Form1

Private Sub Form1\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

MessageBox.Show("Wel come to Rapti Engineering College Ghorahi Dang")

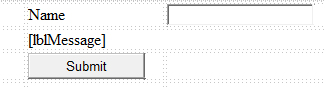
End Sub

End Class

**Developing a Web Application**

ASP.NET is an essential part of the .NET Framework and includes all the services required to develop web applications that involve minimal coding. The different types of ASP.NET applications that can be built in Visual Studio 2010 include Websites and Web applications.

Eg:



Protected Sub btnSubmit\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btnSubmit.Click

lblMessage.Text = "Wel come, " & TextBox1.Text

End Sub

**Working with Visual Basic Operator (Console Application)**

Module Module1

Sub Main()

Dim firstno As Integer

Dim secondno As Integer

Dim message = "Thanks"

Console.Write("Enter First Number :")

firstno = Convert.ToInt16(Console.ReadLine())

Console.Write("Enter Second Number:")

secondno = Integer.Parse(Console.ReadLine())

Console.Clear()

Dim Adds = firstno + secondno

Dim subs = firstno - secondno

Dim muls = firstno \* secondno

Dim divs = firstno / secondno

Console.WriteLine("Addition : " & Adds)

Console.WriteLine("Substraction : " & subs)

Console.WriteLine("Multiplication : " & muls)

Console.WriteLine("division : " & divs)

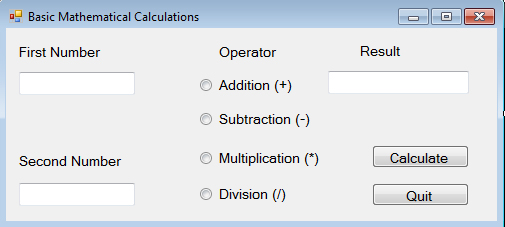
Console.WriteLine(message)

Console.ReadLine()

End Sub

End Module

**Working with Visual Basic Operator (Windows Forms Application)**



Public Class Form1

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

End

End Sub

'declare firstno and secondno varibale

Dim firstno, secondno As Double

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

'assign text box values to variable

firstno = TextBox1.Text

secondno = TextBox2.Text

'Determine checked button and calculate

If RadioButton1.Checked = True Then

TextBox3.Text = firstno + secondno

End If

If RadioButton2.Checked = True Then

TextBox3.Text = firstno - secondno

End If

If RadioButton3.Checked = True Then

TextBox3.Text = firstno \* secondno

End If

If RadioButton4.Checked = True Then

TextBox3.Text = firstno / secondno

End If

End Sub

End Class