UNIT 1

Overview of the Microsoft .NET Platform:

1. Future of computing and the Microsoft's vision
2. Understanding the motivation behind the .NET platform
3. Introduction to .NET

Presented By : Aditya Joshi
Asst. Professor
Department of Computer Application
Graphic Era Deemed to be University
Dated : 12-01-2021

Future of computing and the Microsoft's vision

- Microsoft .NET is a vision to better enable the capability of writing software in simple way that securely connect and interoperate with each other any time, any place and on even in any device.

- .NET is a Microsoft development model in which software becomes platform and device independent and data becomes available over the network.

- .NET is a Microsoft Strategy for connecting systems, information and devices through Web Services to collaborate and communicate effectively.

- Idea is that all devices will some day be connected by a global broadband network and that software will become service provided over this network.

- Few years back, Microsoft had VB & VC++ only to compete against Java. Java being Web-Centric language, was becoming the first choice and a Major cause of migration of programmers from MS to SUN Technologies(Now Oracle).

- To Win back the race from Java, Microsoft put their best men at work for a secret project called Next Generation Windows Services.

- NET, primarily inspired from the ideas of J2EE, has really outperformed their competitors.

NOTE:

To meet the challenge of next generation of Internet-Based Computing, Microsoft has come up with the .NET Platform.

Understanding the motivation behind the .NET platform

The other themes of Application Development are

1. C/Win32 API Programming
2. C++/MFC Programming
3. Visual Basic 6.0 Programming
4. Java/J2EE Programming
5. COM Programming
6. Windows DNA Programming

1. C/Win32 API Programming

- C Programming integrated with windows Application programming interface was the most traditional approach for writing software for the windows family of Operating Systems.
- A numerous applications have been successfully created using this approach.
- C is a very concise language . C developers are forced to contend with manual memory management, ugly pointer arithmetic, ugly syntactical constructs.
- It’s a structured language, so its lacks the benefits of object oriented programming.
- When it combines with thousand of global functions and data types defined by the Win32 API creates a difficult language so there are so many bugs around.

2. C++/MFC Programming

- C++ programming language was a vast improvement over C/Win32 API development.
- C++ can be thought of as an object oriented layer on top of C.
- Thus, though C++ programming benefit from the OOP approaches like polymorphism, encapsulation and inheritance, still there are primarily: lacking memory mgmt., complex pointer arithmetic & syntactical constructs.
- Despite of its complexity, various C++ frameworks exists today.
- MFC (Microsoft foundation class) wraps a complex subset of raw Win32 APIs behind a number of classes, macros and code generation tools (wizards). It facilitates construction of Win32 applications with the help of C++ classes.
- Still C++ programming remains a complex, difficult and error-prone experience, given its historical roots in C.

3. VB 6.0 Programming

- many programmers have shifted away from the world of C(++)-based frameworks to kinder, gentler languages such as Visual Basic 6.0 (VB6).
- VB6 is popular due to its ability to build complex user interfaces, code libraries (e.g., COM servers), and data access logic with minimal effort
- Moreover, VB6 doesn’t provide the ability to build multithreaded applications unless you are willing to drop down to low-level Win32 API calls (which is complex at best and dangerous at worst).

4. Java / J2EE Programming

- The Java programming language is (almost) completely object oriented and has its syntactic roots in C++
- Although Java is a very elegant language, one potential problem is that using Java typically means that you must use Java front-to-back during the development cycle.
- In effect, Java offers little hope of language integration, as this goes against the grain of Java's primary goal (a single programming language for every need).

5. COM Programming

- The Component Object Model (COM) was Microsoft's previous application development framework.
- COM is an architecture that says, "If you build your classes in accordance with the rules of COM, you end up with a block of reusable binary code."
- The beauty of a binary COM server is that it can be accessed in a language independent manner. Thus, C++ programmers can build COM classes that can be used by VB6. Delphi programmers can use COM classes built using C, and so forth.

Unloading COM Components:

COM Objects also require a special logic for freeing up the space from memory, called Reference Counting. When an object's active reference count reaches zero, the object is deleted from the memory. If Circular reference exists between two COM components, they would not be freed from the memory.

Versioning Problem (DLL hell):

Whenever applications that use COM Components were installed on a machine, the installation process would update the system registry with the component's information. There was a chance that these DLLs would be overwritten when some other applications were installed on the same computer. Therefore, an application that had been referring to one particular DLL would start referring to the wrong DLL. This caused a major problem when the application was referring to a particular version of DLL.

6. DNA Programming

- Over the last several years, Microsoft has been adding more Internet - aware features into its family of operating systems and products.
- Sadly, building a web application using COM-based Windows Distributed interNet Applications Architecture (DNA) is also quite complex.
- Some of this complexity is due to the simple fact that Windows DNA requires the use of numerous technologies and languages (ASP, HTML, XML JavaScript VBScript and COM(+) as well as a data access API such as ADO).

Features Provided by .NET

- Full interoperability with existing code.
- Complete and total language integration.
- A common runtime engine shared by all .NET aware languages.
- A base class library.
- No more COM plumbing
- A truly Simplified deployment model.

Introduction to .NET

- .NET is a **software framework** which is designed and developed by **Microsoft**. The first version of the .NET framework was **1.0** which came in the year **2002**. In easy words, it is a virtual machine for compiling and executing programs written in different languages like **C#, VB** etc.
- It is used to develop Form-based applications, Web-based applications, and Web services. There is a variety of programming languages available on the .NET platform, VB. and C# being the most common ones. It is used to build applications for Windows, phone, web, etc. It provides a lot of functionalities and also supports industry standards.
- .NET Framework supports more than **60 programming languages** in which **11 programming languages** are designed and developed by Microsoft. The remaining Non-Microsoft Languages which are supported by .NET Framework but not designed and developed by Microsoft.

11 Programming Languages which are designed and developed by Microsoft are:

- C#.NET
- VB.NET
- C++.NET
- J#.NET
- F#.NET
- JSCRIPT.NET
- WINDOWS POWERSHELL
- IRON RUBY
- IRON PYTHON
- C OMEGA
- ASML(Abstract State Machine Language)