Introduction to .NET Framework
Lecture Outline

• Visual Programming Environment.
• Introduction to .NET
• Introduction to visual studio
• Introduction to visual studio .NET
• .NET Framework.

• Major Components of .NET Framework:
  1. Framework class library (FCL)
  2. Common Language Runtime (CLR)
  3. Common Language Specification (CLS)
• Namespaces
Visual Programming Environment

Environment which allows the use of visual expressions (such as graphics, drawings, animation or icons) in the process of programming.
The technique of Visual Programming allows programmers to create GUI without writing any code.

Microsoft Visual Studio .NET is a visual programming environment.
First: what is Visual Studio??
Second: what is .NET??
1. Visual Studio

- Is the main Integrated Development Environment (IDE) from Microsoft.
- It can be used to develop console and GUI applications.
Integrated Development Environment (IDE)

- In computing, an integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.

- An IDE normally consists of a source code editor, a compiler and/or interpreter, build automation tools, and (usually) a debugger.
Visual Studio Environment

```csharp
doing System.Collections.Generic;
    using System.Collections.Generic;

    {
        public class EstudianteRepository : Repository<EstudianteSE>
        {
            private readonly string _databaseName;

            public EstudianteRepository(string databaseName) : base(databaseName)
            {
                _databaseName = databaseName;
            }

            public EstudianteSE GetEstudianteByMatricula(int matricula)
            {
                IConnectionFactory<int> _connectionFactory = new OrmEstudianteByMatriculasConnectionFactory();
                EstudianteSE _estudianteSE;

                try
                {
                    _estudianteSE = FindOne(_connectionFactory, new OrmEstudianteFactory(), matricula);
                    if (_estudianteSE == null)
                    
                
```
Visual Studio Features

• Visual Studio includes a code editor supporting some features like IntelliSense.
Visual Studio Features

1. **IntelliSense**: shows the available classes and the methods and properties available on those classes.

2. **Designers**: Visual Studio includes visual WYSIWYG designers for GUI and other applications.

3. **Debugging**: is the ability to step through your application line by line as it is executing.

4. **Organization**: it provides intuitive methods for organizing your various code files into projects and your various projects into solutions.
Development of Visual Studio

- Visual Studio 97
- Visual Studio 6.0 (1998)
- Visual Studio .NET (2002) - New programming language that targets .NET
- Visual Studio .NET 2003 - First release to support for developing programs for mobile devices.
- Visual Studio 2005 – Removes .NET only from name.
- Visual Studio 2010 -- better supports multiple document windows and floating tool windows.
Now what is .NET?
Introduction to .NET

• 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.
Introduction to .NET

• The .NET is one over which Web-Based applications can be distributed to variety of devices (such as cell phones) and desktop computers.

• Offers a new programming model that allows programs, created in disparate programming languages, to communicate with each other.

• Additional information available at Microsoft Web site www.microsoft.com/net
You can develop such varied types of applications. That’s fine. But how?
Introduction to .NET

• As with most of the programming languages, .NET has a complete Software Development Kit (SDK) - more commonly referred to as .NET Framework SDK

• It provides classes, interfaces and language compilers necessary to program for .NET.

• Additionally it contains excellent documentation and Quick Start tutorials that help you learn .NET technologies with ease.
Introduction to .NET

.NET Development Tools

• If you are developing applications that require speedy delivery to your customers and features like integration with some version control software then you need some Integrated Development Environment (IDE).

• The new Visual Studio.NET is such an IDE. **VS.NET** is a powerful and flexible IDE that makes developing .NET applications a breeze.
So....

VS + .NET = VS.NET

VS .NET is:
1. A Visual programming environment;
2. That represents the best development environment for the .NET platform.
Introduction to VS .NET

• **VS .NET comes with:**
  1. the .NET Framework,
  2. several programming languages including Visual Basic, Visual C++, and Visual C#.
Overview of the VS.NET IDE

- Integrated Development Environment
  - Visual Basic.NET
    - Many language enhancements
    - Inheritance, Overloading, Free Threading
  - Visual C++
    - Integration with .NET Framework with managed extensions (classes)
  - C#
    - New development language
    - Based on C/C++ with Garbage Collection/Memory Management
  - JUMP (Java User Migration Path) to .NET

http://msdn.microsoft.com/vstudio
Major Components of .NET Framework

- Framework Class Library (FCL)
- Common Language Specification (CLS)
- Common Language Runtime (CLR)
1- Framework class library (*FCL*)

- OO, Pre-packaged classes ready for reuse.
- Used by any .NET language.
- It contains around 3400 classes classified logically into *namespaces*. Each class contains numerous methods and properties which you will use for your programming tasks.
Major Components of .NET Framework

2 - Common Language Runtime (CLR)

Central part of framework: is the virtual machine component of the .NET framework. All .NET programs execute under the supervision of the CLR.

Compilation process: Two compilations take place:

1. Programs compiled to Microsoft Intermediate Language (MSIL)
   - Defines instructions for CLR
2. MSIL code translated into machine code
   - Machine code for a particular platform
CLR: Execution Model

Source code
- VB
- C#
- C++

Managed code

Common Language Runtime

Machine Code

Operating System Services
Why two compilations?

1. **Platform independence** - **Portability**
   - .NET Framework can be installed on different platforms
   - Execute .NET programs without any modifications to code

2. **Language independence** – **Inter-operability**
   - .NET programs not tied to particular language
   - Programs may consist of several .NET-compliant languages
   - Programs written in different languages are all compiled into MSIL—the different parts can be combined to create a single, unified program.
Major Components of .NET Framework

3- Common Language Specification (CLS)

- The CLR allows objects created in one language to be treated as equal citizens by code written in a completely different language.
- To make this possible, Microsoft has defined the CLS that details for compiler vendors the minimum set of features that their compilers must support if they are to target the runtime.
- Any language that conforms to the CLS can run on the CLR.
Major Components of .NET Framework

- VB
- C++
- C#
- JScript
- ... (ellipsis)
- CLS
- ASP.NET: Web Services and Web Forms
- Windows Forms
- ADO.NET: Data and XML
- FCL
- CLR
- Operating System Services

Visual Studio/.NET
As said earlier, FCL contains classes classified logically into *namespaces*.

The .NET Framework is a whole lot of Classes (called *Namespaces*) and the technology to get those Classes to work.
Namespaces

- Most of the built in classes are part of either `System.*` or `Microsoft.*` namespaces.
- It encapsulates a large number of common functions, such as file reading and writing, graphic rendering, and database interaction, among others.
- The .NET class libraries are available to all .NET languages.
Namespaces

- A Namespace is a group of Classes which are grouped together.

- Namespaces are organized hierarchically.

- Namespaces developed by .NET team begin with `System`. 
Namespaces

If we can explore the `System` namespace little bit, we can see it has lot of namespace. *For example:*

- The `System.IO` Namespace groups together Classes that you use to read and write to a file.
- `System.Windows.Forms` Includes classes for creating Windows based forms.
Namespaces

• A single **form** is a Class available to **Forms:**
  
  System.Windows.Forms.Form

• A **Button** is also part of the **Forms** Class:
  

• As too is a **Textbox:**
  
  System.Windows.Forms.TextBox
Namespaces

• Use dot syntax to connect namespaces together:
  – A period separates each namespace name

  System.Drawing
  System.Windows.Forms
  System.Data
Namespaces

Some Namespaces and their use:

- **System**: Includes essential classes and base classes for commonly used data types, events, exceptions and so on
- **System.Data**: Includes classes which lets us handle data from data sources
- **System.Drawing**: Provides access to drawing methods
- **System.Net**: Provides interface to protocols used on the internet
- **System.Web**: Includes classes and interfaces that support browser-server communication
The Organization of Namespaces

System namespace
  System

Drawing namespace
  System.Drawing

Windows.Forms namespace
  System.Windows.Forms

Data namespace
  System.Data

Printing namespace
  System.Drawing.Printing
Lecture Summary

*IT 211* gives an introduction to how to develop .NET software with *Visual C# .NET*. 
SELF-REVIEW EXERCISES

1. **Fill in the blank:**

   1. Programs that translate high-level language programs into machine language are called

      _______________

2. **True or False:**

   1. Visual Basic .NET is the only language available for programming .NET applications.

   2. Computers can directly understand high-level languages.
SELF-REVIEW EXERCISES

3. What is the meaning of the following terminologies:
   - MSIL
   - ADO .NET
   - ASP .NET
   - Portability
   - Inter-operability

4. How many levels of compilation happen in .NET Framework?