Lecture Outline

- Objects and Classes
- Event-Driven programming
- Getting started with VB.NET

What are Objects?

An Object:
- Has a unique identity
- An object has properties (or attributes) which distinguish it from other objects and others of its class – the object’s data
- Has methods (or procedures) which enable it to respond to external events – allow it to interact with other objects, the system, and the user
- Is an instance (example) of a class

Object – General Example

- Consider a butterfly (an object)
- Its distinguishing features (properties):
  - species
  - size
  - age
  - emerging date and time ...
- Its actions within its environment (methods):
  - flying (air)
  - eating (flowers)
  - mating (other butterflies)
  - egg laying (leaf)

Classes and Objects

- A Class describes the basic attributes and operations of a set of objects.
- An Object is an instance or example of the class
  - Property values describe a unique combination
- Example Class: Car
- Example Instances:
  - White Toyota Camry, 5-door, SHS 767
  - Blue Hyundai Excel, 3 door, PLI 200
  - Black Mazda Premacy, 5 door, FLR 687

Attributes or Properties

- Characteristics that describe a particular object
- All objects in a class have an identical set of properties
- The attributes of one object can have different values to the attributes of another object of same class.
- Values of attributes can be changed during the lifetime of an object.
Operations or Methods or behaviour

- Things that objects can do (when asked by program)
- Can modify the object's attributes - mutator
- Can report about object's attributes or state – accessor
- One object can request another object to perform an operation by sending it a message
  - `lightSwitch.TurnOn()` – sends a message to the light switch.

OOP Terminology

- OOP acronym for object-oriented programming
- Object has attributes/properties and behaviour/methods
- Objects are instances of a class (instantiated from a class)
- Class
  - A pattern or blue print used to create an object
  - Every object comes from a class
  - Contains (encapsulates) all the attributes and behaviours that describe the object

OOP Terminology

- Abstraction
  - Refers to the hiding of the internal details of an object from the user
- Inheritance
  - Refers to the fact that you can create one class from another
  - New class is called the derived class
  - New class inherits all the attributes and behaviours of the old class - base class
- Polymorphism
  - Allows the same instruction to be carried out differently depending on the object. E.g: open door, open jar, open eyes, open envelope

What are Events?

- An external trigger to do something.
- Example:
  - When phone rings, you stop what you are doing and answer the phone
  - When you click buttons/icons on the computer screen, the computer then responds by doing something (such as saving your file).

Event-Driven Programming

- Write program that responds to events
- Events are generated by User (at run-time)
  - For Example:
    - User clicks a 'help' button on the screen
    - User selects an item in a list
- Order of events not known in advance
  - See stop watch example on Website

VB .NET Programming

- Visual Basic .NET is an Object-Oriented language
- You use Event-Driven Programming to allow user interaction
- You use some principles of Structured programming (Structure Theorem) to write small pieces of the application.
Some Visual Basic .NET Graphic Objects

- Text box
- List box
- Label
- Button
- Option Button
- Check box
- Forms (Class System.Windows.Forms.Form)

Some Properties of VB.NET Objects

- Name – uniquely identifies the object in the program
- Text – the caption of a button, contents of a text box or label, the title of a form
- Size, Height and Size.Width – the width and height of the object (dimension in pixels)
- Visible – Whether the object should be shown
- Font – the face, style, and size of an object's font
- BackColor, ForeColor – color of background or foreground

Each VB.NET Graphical Object has an extensive list of properties which can be individually set.

Getting Started with VB.NET

- Start MS Visual Studio .NET
- You will get a “Start Page Window”
  - Eg: My Profile
    > allows you to customise various program settings in the IDE (such as keyboard scheme, window layout, and help filter)
    > A collection of customised preferences is called a profile (a set of predefined profiles is available)
    > The default profile is Visual Studio Developer

Visual Studio .NET - Start Page

- Your start screen may look slightly different depending on your version and its configuration.

Starting a New Project

- We will be creating Windows Applications
- They will be Visual Basic Projects
  - Project Files (.vbproj)
    > a document that contains references to all project items such as forms and classes in addition to project references and compilation options.
  - The project will be part of a Visual Studio.NET Solution
    - Solution Files (.sln, .suo)
      > The .sln extension is used for solution files that link one or more projects together, and are also used for storing global information.
      > The .suo file extension is used for Solution User Options files that accompany any solution records and any customisations you make to your solution.
The Visual Studio .NET Integrated Development Environment (IDE)

- **Window’s Form designer window**
  - Design screens
  - Tool bar
- **Code Editor**
  - Specify/implement code
- **Properties window**
  - Change design attributes of the objects on a form
- **Solution Explorer**
  - Shows the Projects and Forms in an application
- **Main Window**
  - At top of screen [title bar, menu bar, standard tool bar]

---

**VB.NET IDE**

- Main window
- Toolbox (auto-hidden)
- Solution Explorer window
- Properties window
- Window’s Form designer window

---

**Reading/Questions**

- **Take a moment to note down**
  - The difference between an object and a class
  - the components of Visual Studio .NET IDE
  - questions you want to ask
- **Reading**
  - Unit Guide, SG 2
  - Zak p1-20

---

**Lecture Outline**

- Recap VB.NET IDE
- VB.NET objects
- Writing VB.NET code

---

**VB.NET**

- Visual Basic.NET is one of several languages supported by Visual Studio (VS)
  - Other languages supported are called C++, C# and Java
- Visual Studio is an IDE
  - IDE = Integrated Development Environment
  - (Your first TLA – three letter acronym)
  - A place where you write code with built in
    - Programmer’s editor
    - GUI (graphical user interface) form developer
    - Syntax checking
    - Context sensitive [F1] help
    - Compiler
VB.NET / Integrated Development Environment contd.

- Toolbox window
  > contains set of controls which make up a VB application.
- Windows Form designer window
  > central part of VB. It provides the display area for the application acts as a container for all the controls
- Properties window
  > Each object in VB has a set of characteristics called properties. This window contains the properties that control the object’s appearance and behavior.
  > Object box, Properties list, Description pane, settings box
- Main window
  > At top of screen. Contains
    - Title bar: displays name of project
    - Menu bar: displays commands used to build VB application
    - Standard toolbar: quick access to commonly used menu commands

Solution Explorer Window

- Displays a list of the projects contained in the current solution
- Each project has:
  > a Reference folder
  > AssemblyInfo.vb file (contains code to deploy application)
  > Form1.vb file (contains code that instantiates objects)

NOTE:
- The reference folder contains references, i.e. addresses of memory cells; each reference points to a namespace. A namespace contains code that defines a group of related classes. E.G.
  > System.Windows.Forms namespace contains the definition of the Windows Form class,
  > is the class used to create a Form object

Working with objects/controls

- form, label, picturebox, button, textbox, listbox
- demonstration
  > Adding a control to a form
  > sizing, moving and deleting a control
  > setting properties of a control at design time
  > changing properties for more than one control

Form Control

- used as a backdrop for other controls
  > useful properties include:
    > Name naming convention = frmSomething
    > Text the text inside the title bar of the form
    > BackColor no need to type code, just click on three dots then choose a colourBox
    > BackgroundImage used to display a graphic as a background
    > Font sets font style, bold, italics, underline etc
    > StartPosition determines where the windows form object is positioned when it first appears on the screen
    > Size used to set the ht and width of the form on the screen

Label control

- Purpose of label control is to display text
  > naming convention = lblSomething
  > useful properties include:
    > FlatStyle 3d or flat
    > BackColor colour of background of label
    > ForeColour colour of text within label
    > BorderStyle none or fixed single (box around text)
    > Text the text on the label
    > Font appearance as applied to the text
    > Size set label’s height and width
    > Position set labels x and y coords on the form
    > AutoSize True or False (True enables label to expand with text)
**PictureBox Control**

**Used to hold an image**

naming convention = picBox

- useful properties include:
  > image: used to select an image to place inside the picBox (click on three dots to open an existing image)
  > SizeMode: set whether the chosen picture fills the entire space of the image control
  > Size: used to set the size of the image
  > Location: used to set the positioning of the image

**Text Box Control**

**Used to get input such as text from the user**

naming convention: txt

- useful properties include:
  > BorderStyle: Can have a box around text
  > Enabled: True or False (False means user cannot enter text into text box)
  > Font: same as before
  > Size, Location: same as before
  > MultiLine: True or False (True lets user hit return and keep typing)
  > ScrollBars: none, vertical, horizontal, or both (requires MultiLine to be True)
  > Text: displays initial (default) text in text box; used to lighten burden on user

Always add a label next to a text box to guide the user in their input.

**List Box Control**

**Used to display items in a list box**

naming convention: lst

- useful properties include:
  > Items: items into your listBox
  > Enabled: True or False (False means user cannot enter text into text box)
  > Font: same as before
  > Size, Location: same as before
  > ScrollAlwaysVisible: True or False

- Useful methods
  > Clear(): clears all items in a list box
  > Add(...): appends an item to the list box

**Button Control**

**used to allow user to initiate some coded behaviour**

naming conventions cmd

- useful properties include:
  > Text: the text on the button
  > Font: font appearance as applied to the Caption
  > Size: used to set the size of the button on the screen
  > Location: used to set the positioning of the button on the screen
  > Enabled: True or False

**Access Keys**

- Reduce the use of the mouse, so enable speed for user input
- Add an ampersand (&) in front of access key letter in the caption
  
  E&xit as the caption on a command button lets the user select the button by hitting Alt plus x simultaneously
- Important for user friendliness

**VB.NET Object Properties**

- Qualities or attributes of an object/graphic-control
- Can be changed by programmer
  - design time – in the ‘properties’ box.
  - Can be changed by user
    - run time – e.g. typing into a text box
- Dynamically (in code):
  
  ObjectName.PropertyName = Value

  - TextBox1.Text = “Hello World”
  - TextBox1.Font.Bold = True
Properties are assigned by ...

- **Properties Window**
  - Click object with mouse – updates RHS panel
  - If not visible, press ‘F4’
- **Code associated with an event**
  - Typed in the code-window
  - e.g. Text1.Visible = True
  - e.g. Button1.Enabled = False

Objects and Events

- **Example of objects**
  - Form, pictureBox, label, command buttons
- **Example of events (user actions)**
  - click, double click, scroll, keypress
- **Event procedure (method)**
  - VB.NET instruction, or code that tells an object how to respond to an event

Starting and Ending an Application

- **To run an application**
  - Click Debug then click start or press F5
  - Visual Studio .NET creates an executable file(.exe) that can be run outside the IDE

  - Demo:
    > Lets run our demo application. What happens?

VB.NET IDE

- **Code Editor Window**
  - The Class Name list box lists the names of the objects included in the user interface
  - The Method Name list box, on the other hand, lists the events to which the selected object is capable of responding
  - You use the Class Name and Method Name list boxes to select the object and event, respectively, that you want to code
  - To help you follow the rules of the Visual Basic .NET programming language, called syntax, the Code Editor provides you with a code template for every event procedure

Writing Code - Syntax and Semantics

- **Programming involves understanding the Syntax and Semantics of a particular programming language, to form Statements that tell the computer what to do.**
- **Statement**
  - An instruction to the computer (specifically, for the compiler)
- **Syntax**
  - The rules for constructing valid statements
- **Semantics**
  - The meaning or interpretation of a statement

Writing Code – Organisation of Statements

- **Statements are organised into a source-code file**
- **The source code file is divided into sections:**
  - Class definition
  - Form Layout information
  - Global Variables and Constants
  - New data type
  - Procedures and Functions
### VB.NET Form – Class Definition

- A class definition is simply a block of code that specifies (or defines) the attributes and behaviors of an object
  - When you create a Form, VB.NET produces the relevant code to describe the layout of objects on the Form
  - This code is hidden in the 'Windows Form Designer Generated Code' region.
- When you start the application, Visual Basic .NET uses the class definition to create the object
  - Your Form is displayed

### Event Procedures

- Actions—such as clicking, double-clicking, and scrolling—are called events
- The set of Visual Basic .NET instructions, or code, that tells an object how to respond to an event is called an event procedure
- You write statements inside an event procedure to specify the individual steps required for the input, processing, and output.
- To help you follow the syntax rules of the Visual Basic .NET programming language, the Code Editor provides you with a code template for every event procedure

### Syntax of an Event Procedure

<table>
<thead>
<tr>
<th>Start of a Procedure</th>
<th>Name of subroutine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Private Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click</td>
<td>Name of subroutine</td>
</tr>
<tr>
<td>Textbox1.Text = &quot;Hello&quot;</td>
<td>Assignment statement - Changes the 'text' property</td>
</tr>
<tr>
<td>End Sub</td>
<td>Event to respond to</td>
</tr>
<tr>
<td>End of a Procedure</td>
<td></td>
</tr>
</tbody>
</table>

### Writing Visual Basic .NET Code

- The first line in the Code template is called the procedure header
- And the last line is called the procedure footer
- A keyword is a word that has a special meaning in a programming language
- The Sub keyword is an abbreviation of the term sub procedure, which, in programming terminology, refers to a block of code that performs a specific task
- The Private keyword indicates that the procedure can be used only within the class in which it is defined

### Writing VB.NET code

- VB.NET provides Code Template for each Event Procedure
- Example:
  ```vbnet
  Public Class Form1
  Inherits System.Windows.Forms.Form
  Private Sub btnExit_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles btnExit.Click
  Close()
  End Sub
  End Class
  ```
- Keywords = Private Sub, End Sub
- Object Name = btnExit
- Event = Click ()

### Writing Vb.NET code

- Close() [or Me.Close()] Method
  - Terminates the current application
  - The Me in the instruction refers to the current form
  - The Close is the method
Is VB.NET Interpreted or Compiled?

- Aargh…. It’s both
  - Your VB.NET code is first semi-compiled into something called Microsoft Intermediate Language (MIL) and stored in a file with the suffix .EXE
  - At run time, the MIL is interpreted by the Microsoft Common Language Runtime (CLR) program and converted into machine code

End - Today we covered

- More Objects and the Click Event
- Managing a VB.NET application
- Coding a VB.NET application
- How to create, save, run and stop a VB application

Reading/Questions

- Take a moment to note down
  - The VB objects you remember and their purpose
  - the layout and components of VB.NET IDE
  - questions you want to ask
- Reading
  - Zakp20-40
  - Robertson Chpt 11
  - Unit Book, SG 2