COSC 1530

Introduction to Computer Use II: Programming

Winter 2005 (Section M)

Topic A: Introduction to Problem Solving and Visual Basic

Monday, January 9 2006

Bill Kapralos

Overview (1):

- **Before We Begin**
  - Some administrative details
  - Some questions to consider
- **Object Oriented Programming (cont.)**
  - Fundamental concepts
    - Shape Example
- **Introduction to Programming with VB 6.0**
  - MS Visual Basic 6.0
  - The VB Integrated Development Environment (IDE)
  - "Live" examples

Before We Begin

Administrative Details (1):

- **Some Reminders**
  - Office hours: MW 2:30-3:30pm in CSE 2015
  - Course URL (where the lecture notes can be found)
    http://www.cs.yorku.ca/~billk/cosc1530/
- **Glade Lab Accounts**
  - If you don't have an account, go to the Computing Commons Lab in the William Small Center (opposite the Chemistry building by Tim Horton's) and speak to the lab attendant

Administrative Details (2):

- **Lab Exercises**
  - Remember that lab exercises are specified in the course outline (essentially one per week)
    - Due the following Monday → for example, Ex. 2-3 is intended to be worked on during week 2 (week of Jan. 9) and handed in the following Monday (Jan. 16) by 12:00 noon!
    - Should be working on Lab 2-3 this week

Some Questions to Consider (1):

- What is a high-level programming language ?
- What is a low-level programming language ?
- What is a procedural programming language?
- What is an object oriented programming language ?
- What is an event ?
- What is an event driven programming language ?

Object Oriented Programming

Fundamental Concepts (1):

- **Central Concepts to OOP**
  - Object → an entity that you can manipulate in your program generally by calling methods associated with the object
    - Think of it as a "thing" (e.g., a noun)
    - Think of an object as a "black box" with a public interface (methods you can call) and a hidden implementation (the code and data that are necessary to make these methods work)
    - As a user of the object, you don't have to worry about the implementation details!

Fundamental Concepts (2):

- **Central Concepts to OOP (cont.)**
  - Class → defines the methods (e.g., sub-programs that perform (compute) a particular task) and any properties for the objects
    - Every object belongs to a class → every object is an instance of a particular class
    - Classes are "factories" for objects (e.g., used to generate objects)
    - Can have many objects of a particular class and the properties of each object may have different values

Fundamental Concepts (3):

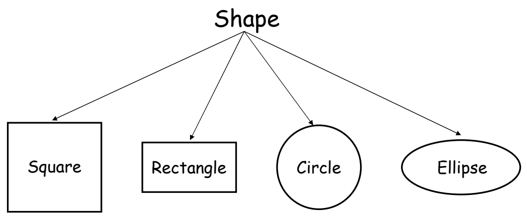
- **Central Concepts to OOP (cont.)**
  - Inheritance → a mechanism for enhancing existing classes
    - Suppose you need to implement (create) a new class and a class representing a more general concept is available, then the new class can use (inherent) from the existing class
    - Suppose you have a class called "Shape" but you need to define a "Rectangle" → a rectangle is itself a shape albeit a specific shape!

Fundamental Concepts (4):

- **Central Concepts to OOP (cont.)**
  - Properties → tell us something about an object such as its name, color, size, location or how it behaves
    - Think of properties as "adjectives" that describe objects (which have been defined as "nouns")
  - Methods → associated with objects
    - Think of methods as "verbs"
    - Perform a specific task
    - For example, in Shape class, we may have a method called "getArea" that returns the area of the shape object → we don't know how it's calculated but we know we get the area (abstraction)

Fundamental Concepts (5):

- **Putting It All Together**
  - An example → shape class



Fundamental Concepts (6):

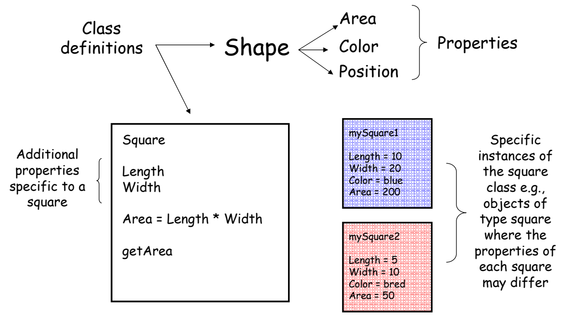
- **Putting It All Together (cont.)**
  - Shape class is the more general definition and then we have specific shapes such as squares, circles etc.
    - Each shape shares certain general properties such as area but calculating area is specific to the type of shape being considered (e.g., calculating area for circle is different than calculating area for square)
    - Each shape can inherent the area property from the general Shape class but then define a method that will calculate the specific area for the shape
    - Can have many instances of a particular shape

Fundamental Concepts (7):

- **Putting It All Together (cont.)**
  - Keep in mind that OOP is a very big topic → we can spend an entire course on OOP
    - In this course we are not going to any great details regarding OOP but we should just be aware of some of the main concepts since VB is itself an OOP language e.g., have an idea of what an object is and what methods and properties are
    - More on this as we work with VB

Fundamental Concepts (8):

- **Putting It All Together (cont.)**



Introduction to Programming with MS Visual Basic 6.0

Microsoft's Visual Basic (1):

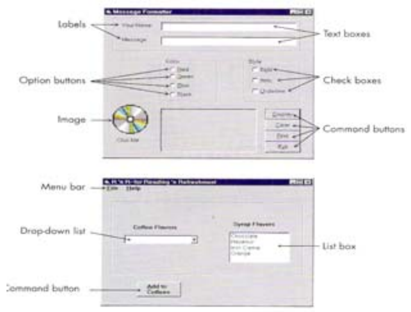
- **Introduction (cont.)**
  - In this course we will be using Visual Basic (VB) 6.0 despite the fact that there is a newer version available (VB.Net)
    - VB.Net is a more complete OOP language however, it is also a more complex development environment and many of its features are beyond the scope of this course
    - VB 6.0 is better suited for the purposes of this course
    - Once you become familiar with VB 6.0, then VB.Net should be easy to pick up

Microsoft's Visual Basic (2):

- **Introduction**
  - As mentioned, an OOP event driven language
  - An integrated development environment (IDE) that runs in the Microsoft Windows environment
  - The "projects" you create will look and act as standard Windows programs
  - Visual Basic (VB) provides the tools you need to create windows or Graphical User Interfaces (GUIs) known as forms in Visual Basic with familiar elements
    - Menus, text boxes, command buttons, scroll bars…

Microsoft's Visual Basic (3):

- **Introduction (cont.)**
  - Example windows (GUIs) with common "elements"



Microsoft's Visual Basic (4):

- **MS Windows Graphical User Interface (GUI)**
  - Defines how various elements look and function
    - Consistent regardless of the application program otherwise, it would be very confusing to MS Windows users!
    - For example, in any Windows application, you know that to exit, you click the "x" in the top right corner of the window
  - With VB, you can create such interfaces!
    - The windows you create in VB are called forms
    - Elements within a window are called controls

Getting Started (1):

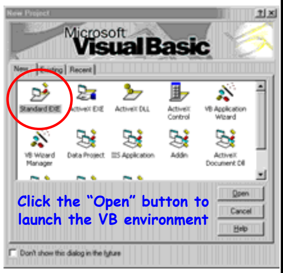
- **Visual Basic Projects**
  - Visual basic makes use of projects → every application/program you create is part of a project
  - A project is a collection of at least two and usually more files
    - The two files we will be most concerned with (at least initially) are project files and form files

Getting Started (2):

- **Visual Basic Project Files**
  - The project file (.vbp extension)
    - A text file that holds the name of the other files in the project and information regarding the project environment
  - The form file (.frm extension)
    - Each form (window) in your project is saved in a separate file
    - Initially, your project will contain one form only but you may add additional forms as you please
    - Holds a description of all the objects (elements) your form contains

Getting Started (3):

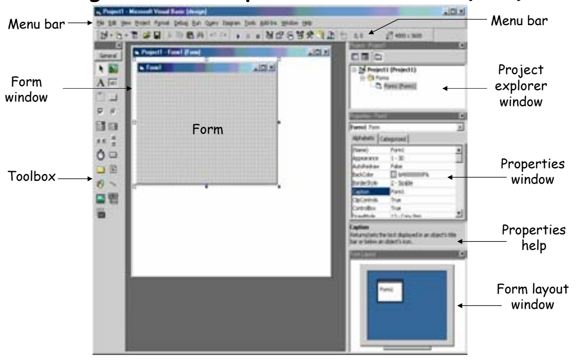
- **So How Do We Start With VB ?**
  - Startup VB 6.0
  - The following window will then appear giving you the option of creating an new project or open an existing one



  - We will only deal with "Standard EXE" projects
    - Although other project types are available, it is beyond the scope of this course to cover them!

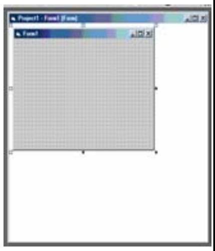
Visual Basic Environment (1):

- **Integrated Development Environment (IDE)**



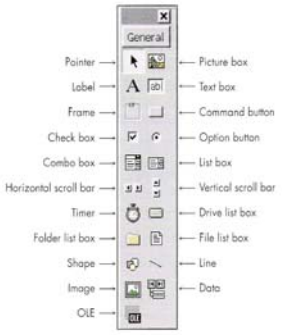
Visual Basic Environment (2):

- **The Form Window**
  - Where you design the forms (windows) that comprise your user interface
    - Typically place objects (e.g., buttons, text boxes etc.) from toolbox on the form
    - When you begin a new project, you are given one default form called Form1
    - When you save the project you will give it a new name



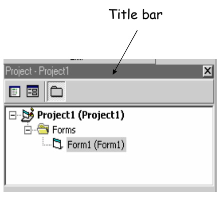
Visual Basic Environment (3):

- **The Toolbox**
  - Holds the tools you use to place controls in a form
    - Labels
    - Command buttons
    - Text box
    - Check box
    - etc.



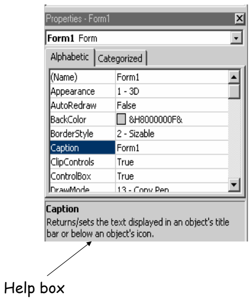
Visual Basic Environment (4):

- **The Project Explorer Window**
  - Holds the file names of the files included in your project
  - The title bar holds the name of the project
  - "Project1" when you first start VB without saving it



Visual Basic Environment (5):

- **The Properties Window**
  - Use this window to set the properties for the objects in your project
  - Includes the "properties help" that gives a description of each property you highlight



(Name)	Form1
Appearance	1 - 3D
AutoRedraw	False
BackColor	&H8000000F&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen

Caption
Returns/sets the text displayed in an object's title bar or below an object's icon.

Finally… (1):

- **"Live" Demo of Visual Basic**
  - Getting started
    - Creating a project
    - The VB IDE and its various components
    - Saving the project
    - We will continue our live demo Wednesday and Friday → we will discuss the examples from Chapter 2 in detail