Test 1 will be Thursday, October 1, in class. It will be closed book and notes, except for one 8.5” x 11” sheet of paper (front and back) with notes. The test will be mostly multiple choice, true/false, and matching type of questions.

Review topics for Test 1 are:

**Chapter 1.**
- Major hardware components and their purpose: CPU, main memory, secondary storage devices (e.g., hard disk), input devices, and output devices
- General category of software: operating system and application software
- Primary methods of programming today are procedural and object-oriented (e.g., Visual Basic)
- The advent of graphical user interfaces (GUIs) has influenced the shift from procedural to object-oriented programming.
- Event-driven programs respond to events or actions that take place while the program is running.
- Visual Basic has many types for controls: Labels, TextBoxes, Buttons, CheckBoxes, RadioButtons, ListBoxes, ComboBoxes, and scroll bars.
- Each control has a name that allows programming statements to access and manipulate it.
- When a control is created it receives a default name (e.g., Label1, TextBox2, etc.). Programmers should change the name of any control that is used in a programming statement to a meaningful name (e.g., lblSalesTotal, txtQuantityPurchased, ...)
- The general steps for developing a Visual Basic application include:
  - **Understanding the problem**
  1. Clearly define what the application is to do: purpose, inputs, processing, outputs
  - **Design screens**
  2. Visualize application running and design its user interface – sketch it on paper
  3. Make a list of the controls needed, decide on their names, and write a description
  4. Define the values of each control’s relevant properties
  - **Design actions behind buttons**
  5. Make a list of methods needed for each control (user actions they must respond to)
  6. Create a flowchart or write pseudocode for each method
  7. Check the code for errors by walking through its execution (sketch screens and trace variables)
  - **Build screens**
  8. Start Visual Studio and create forms and other controls
  - **Write code for actions**
  9. Write code for event procedures and other methods of step 5
  - **Test correctness**
  10. Attempt to run the application and correct syntax errors

Repeatedly run the program with test data as input and correct any run-time or logical errors

**Chapter 2.**
- Comments are started by an apostrophe (‘) or REM and explain the code for the programmer.
- Me.Close( ) statement causes the current form to close.
- An assignment statement can be used to modify the Text property of a control at run-time, e.g., lbl.Text = “new text for the label”
- Click event procedures can be defined for Buttons, PictureBoxes, and Labels.
• VB checks for syntax errors (i.e., the incorrect use of a programming element--keyword, operator, or user-defined name) while you type code and at compile time. The compiler uses a jagged, blue, underline to highlight the syntax error.
• Run-time errors occur when the application is executed (Debug | Start Debugging) and VB is asked to perform an operation that it cannot (e.g., division by zero, convert a string containing letters to an integer, etc.).
• Logic errors occur if your program is syntactically correct and does not cause a run-time error, but the program generates incorrect output due to programmer mistakes.

Chapter 3.
• A TextBox control allow users to type a String of words and characters into its Text property
• String concatenation is performed by the ‘&’ operator
• The control that has the focus receives the user’s keyboard or mouse clicks. The focus can be moved in code by calling the Focus method. At run-time the user can change the focus by Tabbing through all the controls whose TabStop property is True.
• The Tab order can be set by the programmer by change the controls TabIndex properties using the View | Tab Order menu.
• Access key shortcuts can be assigned to Buttons by placing an ampersand (&) before a letter in the Text property
• Forms have an AcceptButton property that refers to the control that receives a Click event when a user presses the <Enter> key.
• Forms have a CancelButton property that refers to the control that receives a Click event when a user presses the <Esc> key.
• A variable is a named spot in memory used to hold information while the program is running
• When a programmer creates a variable using a dimension statement (Dim), the type of data the variable can hold is specified. (e.g., String, Integer, Single, Double, Decimal, Date, Boolean)
• Local variables are defined inside of a procedure and exist only while that procedure is executing. Their scope (where they can be used) is only within the procedure.
• Class variables are defined within a class, but outside of any procedure. Their scope extends across the whole class.
• Calculations are performed by operator similar to algebra. Some of the stranger ones are: ‘\’ for integer division, ‘*’ for multiplication, ‘^’ for exponentiation, ‘MOD’ for the remainder of integer division.
• Operator precedence determines the order of operations with parentheses being used to force some operations to be performed before others.
• VB implicitly tries to convert the value being assigned to the data type of the destination variable (i.e., the variable on the left-hand side of the assignment ‘=’ operator.)
• When Option Strict is On, only widening conversion are permitted where a smaller type is assigned to a variable of a larger data type (e.g., Double variable assigned an integer value)
• Either a type conversion or mismatch run-time error will occur when an automatic conversion is not possible.
• Programmers can use explicit type conversion functions built-in to VB: CDbl, CInt, CStr, CDate, Val
- The ToString methods exist for numeric types to convert to a String for displaying. An optional formatting parameter can specify how a value should be displayed: “f” for fixed-point, “e” for exponential, “c” for currency, “p” for percentage.
- Exception handling using a Try-Catch block followed by an End Try allows user code to be invoked when a run-time error occurs.
- GroupBoxes allow rectangular borders to contain other controls. GroupBoxes aid the user by visually showing related controls.

**Chapter 4.**
- VB allows several decision structures (If ... Then, If ... Then ... Else, If ... Then ... ElseIf, Select ... Case) that allow some statements only to be executed under certain circumstances.
- Relational operators (>, <, =, >=, <=, <> ) can be used to compare two numeric values, or two string values to get a Boolean result (True or False).
- Logical operators (And, Or, AndAlso, OrElse, Xor) can be used to combine two or more Boolean results. The Not operator can negate a Boolean result.
- String function IsNumeric checks to see if the string contains a number
- String method Substring extracts part of a string
- String method IndexOf searches for the location of a substring
- MessageBoxes.Show method can be used to pop up a message box to display some information to the user.
- Input Validation using the TryParse method
- A group of RadioButtons allow the user to select one of several options.
- A group of Check Boxes allow the user to select as many options as they want.