1. Basic Knowledge

1a. [4 points, one point each] Define/explain any **four** of the following terms.

Algorithm:

- A set of step-by-step instructions that when completed, solve a problem.

Procedural programming:

- The program itself specifies the exact sequence of all operations
- Programming logic determines the next instruction to execute in response to conditions and user requests

Object oriented programming (OOP):

- A computer program may be seen as composed of a collection of individual units, or objects, that act on each other, as opposed to a traditional view in which a program may be seen as a collection of functions or procedures or simply as a list of instructions to the computer
- More flexibility, changes to programs can be easily made and is widely popular in large scale software engineering
- Allows for re-use of code

Constant:

- A variable whose value is set upon declaration and cannot be changed at any other point of program execution.

Function:

- A "sub-program" that performs a specific task. User provides zero or more arguments, some operation is performed and a value is returned.

1b. [10 points, 1 point each] **True / False**. For each question, circle your choice of either True or False (do not circle both!).

1) An object's properties can only be changed in design mode.
True ***False***

2) The value of a variable declared locally in one method (e.g., event handler) can be accessed and changed within another method.
True ***False***

3) The number of bytes to represent a variable of type **Integer** is four.
True ***False***

4) All objects of a particular class have the same properties, but the values of those properties may be different.
True False

5) When calling a function, you must always provide at least one argument.
True ***False***

6) A name used for a variable declared locally within one event handler may not be used again for a variable declared in another event handler.
True ***False***

7) A **high-level** programming language provides little or no abstraction from a computer's microprocessor.
True ***False***

8) Visual basic's random number generator function ("**Rnd**") returns a random number between 0 and 100.
True ***False***

9) **Object Explicit** forces all variables to be declared as global.
True ***False***

10) The return value of a function can be assigned to an object's property.
True False

1c. [3 points] List three **advantages** of Object Oriented Programming.

- OOP is easier to learn for those new to computer programming than previous approaches
- Software (code) is often simpler to develop and to maintain, lending itself to more direct analysis, coding, and understanding of complex situations and procedures than other programming methods.
- Promotes abstraction.
- Inheritence.

1d. [2 points] Describe/explain how we can determine whether a variable should be declared **global** or **local**.

- Think about the following: "in how many methods does the variable need to be accessed in". If it is only used within a single method, then the variable can probably be declared local. if it is used in more than one method, it most probably needs tobe declared gloabal.

2. Programming

2a. [2 points] Assume **txtValue** is the name of a Textbox object placed on some Form object. Is the following code segment valid? Explain your answer.

```
txtValue = "100"
```

- Yes, it is valid.
- Recall that each control object has a default property that is assumed when not explicitly specified. In the case of the TextBox, the default property is the Text property. Therefore, above statement is equivalent to: **txtValue.Text = "100".**

2b. [2 points] Assume **text1**, **text2** and **result** are Textboxes placed on a form. List the output displayed in the **result** Textbox after the following code segment is executed. Explain your answer.

```
text1.Text = "100.5"
text2.Text = "45"
result.Text = text1.Text + text2.Text
```

- Recall that the "+" operation for strings is catenation therefore, the above statement will simply catenate the two strings: **100.545**

2c. [2 points] Is the following code segment valid? Explain your answer. You can assume that **CheckIt** is a button placed on a Form object.

```
Const maxValue As Integer = 10

Private Sub CheckIt_Click()
   maxValue = 10 * 2
End Sub
```

- No, it is not valid because maxValue is declared as a constant (e.g., "Const") and therefore, its value cannot be changed at any point of program execution.

2d. [2 points] Is the following code segment valid? Explain your answer. You can assume that **CheckIt** is a button placed on a Form object.

```
Option Explicit
Dim myValue As Integer

Private Sub CheckIt_Click()
   newValue = 15
   myValue = factor * factor
End Sub
```

- No, the above code segment is not valid.
- The "Option Explicit" statement is present indicating that all variables must be declared yet, the variable "newValue" has not been declared

2f. [2 points] Is the following code segment valid? Explain your answer.

```
Dim var1 As Boolean
Dim var2 As Boolean
Dim var3 As Boolean

var3 = var2 NOT var3
```

- No, the code segment is not valid because the "NOT" operator is a unary operator not a binary operator (e.g., it does not take two arguments, only one).

2e. [2 points] Assume **myForm** is a Form object, **CheckIt** is a button placed on **myForm** and **txtAnswer** is a Textbox object placed on **myForm**. List the output displayed in the **txtAnswer** Textbox after the following code segment is executed and explain your answer.

```
Dim x As Double

Private Sub myForm_Load()
   x = 100
End Sub

Private Sub CheckIt_Click()
   Dim x As Double
   x = 200
   answer.Text = CStr(x)
End Sub
```

Output: x = 200

- The variable "x" is a global variable and its value is assigned 100 in the "Load" method. Yet, it is assigned the value 200 in the "Click" method. In both methods, it is the same variable that is accessed therefore, "x=200" is the latest value assigned to x before being displayed.

2f. [2 points] Consider the code segment shown below. What will the value of the variable called **result** be after the code segment is executed ?

```
Dim value1 As Single
Dim value2 As Single
Dim value3 As Single
Dim result As Boolean

value1 = 15.5
value2 = 15.0
value3 = 14.5

result = (value1 < value2) XOR (value3 < value2)
```

Output: result = True

What will the value of the variable called **result** be if we change the type of the variable called **result** to Single as shown below ?

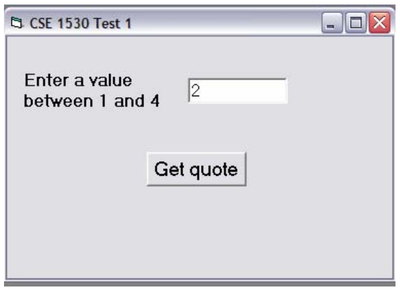
```
Dim value1 As Boolean
Dim value2 As Single
Dim result As Single
Dim result As Boolean

value1 = TRUE
value2 = 15.0
value3 = 14.5

result = (NOT value1) AND (value2 < value3)
```

Output: result = False

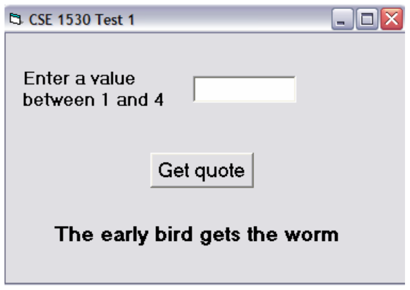
2g. [7 points] Consider a program where the user will enter a number between 1 and 4 in a Textbox object (called **txtValue**) placed on a Form object (called **myForm**) and when a button (also placed on the Form object and called **btnGetQuote**) is pressed, depending on the value entered by the user, one of the four messages (listed below) will be displayed in a Label object called **lblMessage** also placed on **myForm**.

Value Entered by User	Message Displayed
1	"A penny saved is a penny earned"
2	"The early bird gets the worm"
3	"Better safe than sorry"
4	"An ounce of prevention is worth a pound of cure"

Your task is to provide the code for the **Click()** event handler of the **btnGetQuote** button that will implement the above specification. You can **assume** that the user input contains numeric values only (e.g., no need to validate input), however, if the user enters any values other than 1, 2, 3 or 4 you should display the following message also in the lblMessage Label "Only numbers 1,2,3 or 4 allowed". In addition, the **txtValue** Textbox should be cleared after the quote is displayed. Below are two sample screenshots of the form prior to pressing the button and after pressing the button.



Form prior to pressing the "Get Quote" button.



The early bird gets the worm

Form after pressing the "Get Quote" button.

```
Private Sub CheckIt_Click()

      Dim inputValue As Integer
      inputValue = CInt(txtInput.Text)

      If (inputValue = 1)
         lblMessage.Caption = "A penny saved is a penny earned"
      Else If (inputValue = 2)
         lblMessage.Caption = "The early bird gets the worm"
      Else If (inputValue = 3)
         lblMessage.Caption = "Better safe than sorry"
      Else If (inputValue = 4)
         lblMessage.Caption = "An ounce of prevention is worth…"
      Else
         lblMessage.Caption = "Only numbers 1,2,3 and 4 allowed"

      txtInput.text = vbNullString

End Sub
```

Additional Space

Use this page for any additional space you require. Please state question numbers.