



## GEG 4301 Advanced Geomatics

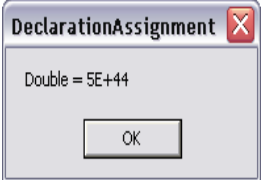
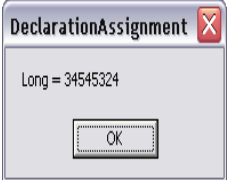
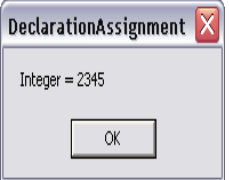
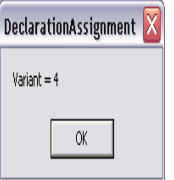
1. [3 points] For the following values, assign each the appropriate VB data type. You must choose the most efficient.

<b>Value</b>	<b>Datatype</b>	<b>Bytes</b>
<b>0</b>	<b>Byte</b>	<b>1</b>
<b>1.235</b>	<b>Single</b>	<b>4</b>
<b>500000</b>	<b>Long Integer</b>	<b>4</b>
<b>"Hello"</b>	<b>String</b>	<b>10+length</b>
<b>Values ranging from 0 to 255</b>	<b>Byte</b>	<b>1</b>
<b>The value 3442 or value "Chester"</b>	<b>String</b>	<b>10+length</b>
<b>32767</b>	<b>Integer</b>	<b>2</b>
<b>False</b>	<b>Boolean</b>	<b>2</b>
<b>45000</b>	<b>Long Integer</b>	<b>4</b>
<b>2000000</b>	<b>Long Integer</b>	<b>4</b>
<b>True</b>	<b>Boolean</b>	<b>2</b>

**A: The efficiency is measured by the number of Bytes that is necessary to use. For example, zero could be Integer using 2 bytes but also Single using 4 bytes, but the most efficient is Byte data type that uses only 1 byte.**

2. [1 point] Provide an example of variable declarations and assignments for a Double, Long, Short, Integer and Variant data types.

**A: Short data type is not supported by VBA. Other data types are given below:**

CODE/VARIABLE	Double	Long	Integer	Variant
<b>Subroutine Title</b>	Sub DeclarationAssign ment()	Sub DeclarationAssign ment()	Sub DeclarationAssign ment()	Sub Declaration Assignment ( )
<b>Variable Declaration</b>	Dim doubleX As Double	Dim longY As Long	Dim integerZ As Integer	Dim variantV
<b>Variable Assignment</b>	doubleX = 5E+44	longY = 34545324	integerZ = 2345	variantV = 4
<b>Message box text</b>	msgbox "Double=" & doubleX, Title:="DeclarationAssi gnment" End Sub	msgbox "Long=" & longY, Title:="DeclarationA ssignment" End Sub	msgbox "Integer=" & integerZ, Title:="DeclarationA ssignment" End Sub	msgbox "Variant=" & variantV, Title:="Declara tionAssignment " End Sub
<b>Output</b>				

3. [1 point] What is the default datatype in VBA?

**A: When the datatype is not specified, *Variant* is the default datatype used in VBA. It can hold data of any type except fixed-length string data. This variable is initially empty which can cause the problems later in programming, thus it is a good programming practice to declare all variables.**

4. [1 point] What is the statement "option explicit" used for in VB? Why is it important.

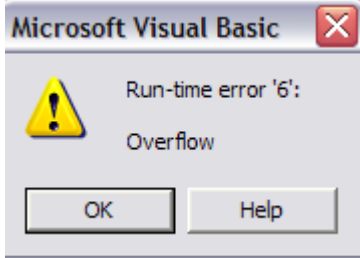
**A: The statement "option explicit" enforces declaration for every and each variable that is used in the program. Once this statement is added to a module it is not possible to use a variable unless it is declared. It is recommendable to set this option permanently under Tools/Options and select Require Variable Declaration. The statement "option explicit" helps to avoid incorrectly typing the name of an existing variable or to avoid confusion in code where the scope of the variable is not clear.**

5. [1 point] Write the shell (wrapper lines) of a public and private subroutine.

Wrapper Lines for a Public Subroutine	Wrapper lines for a private subroutine
<b>Public Sub</b> <i>Title (arguments)</i> <i>statement</i> <b>End Sub</b>	<b>Private Sub</b> <i>Title (arguments)</i> <i>statement</i> <b>End Sub</b>

For questions 6-11, use the "msgbox" function to return displayed values.

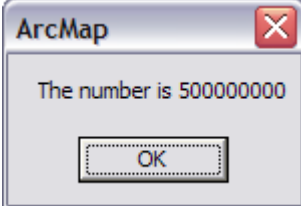
6. [1 point] Create a sub procedure that declares a variable of type Long and assign this variable the numeric value of 5000000000.45 Run the procedure. What happens? Why?

Routine	Output
<pre>Sub Calculus2()   Dim number As Long   number = 5000000000.45   msgbox "The number is " &amp; number End Sub</pre>	

**A: This subroutine could not be executed. MVB error message widow appeared on the screen: Run-time error '6': Overflow. This has happened due to the fact that the variable was declared as Long. Long takes argument in the following range:-2,147,483,648 to 2,147,483,647. The number 5000000000.45 is outside of the range – therefore Overflow message was flagged.**

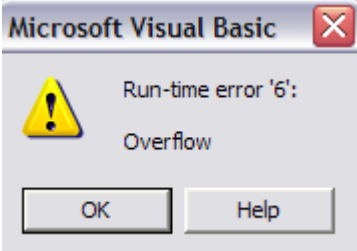
7. [1 point] Repeat the previous question but assign the numeric value of 5000000000.232 to the variable that was declared as a Long. Run the procedure. What happens? Why?

**A: This routine was executed however the number was rounded to 500000000. This is due to the fact that for the variable type Long fractions are rounded. Therefore, the decimal places will not be shown when Long is used.**

Routine	Output
<pre>Sub Calculus2()   Dim number As Long   number = 500000000.232   msgbox "The number is " &amp; number End Sub</pre>	

8. **[1 point]** Create a sub procedure that declares a variable of type Double and assign this variable the numeric value of  $1E+2994$ . Run the procedure. What happens? Why?

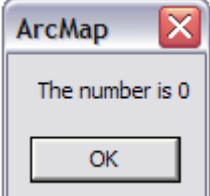
**A:**

Routine	Output
<pre>Sub Calculus2()   Dim number As Double   number = 1E+2994   msgbox "The number is " &amp; number End Sub</pre>	

**The run time error '6: Overflow window will appear. This indicates that the range value of the argument for Double (upper positive value 1.79769313486232E308) was exceeded.**

9. **[1 point]** Create a sub procedure that declares a variable of type Double and assign this variable the numeric value of  $1E-2994$ . Run the procedure. What happens? Why? What happens to your code in this case?

**A:**


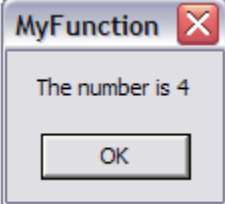
Routine	Output
<pre>Sub Calculus2()   Dim number As Double   number = 1E-2994   msgbox "The number is " &amp; number End Sub</pre>	

**This code was executed and it returns value 0. In this case, the original code was changed to:**

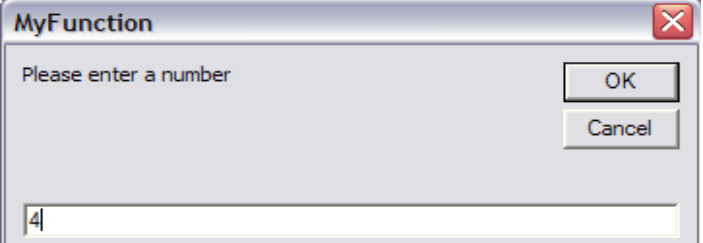
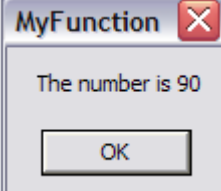
```
Sub Calculus2()
    Dim number As Double
    number = 0#
    msgbox "The number is " & number
End Sub
```

**The range value of the argument for Double was exceeded and because it is very small number the returned value is rounded to zero. The symbol # is a type declaration character for datatype Double.**

10. [1 point] Create a function that is called from a sub procedure and accepts as an argument a number and returns the number unchanged.

Function	<pre>Function MyFunction(Argument)     MyFunction = Argument End Function</pre>
Subroutine	<pre>Sub UsingFunction()     MyNumber = InputBox("Please enter a number",         Title:="MyFunction")     msgbox "The number is " &amp; MyFunction(MyNumber),         Title:="MyFunction" End Sub</pre>
Input Window	
Output	

11. [1 point] Repeat Q#10 but have the function add the number 5 and multiply the result by 10 before being returned.

Function	<pre>Function MyFunction(Argument) MyFunction = (Argument+5) * 10 End Function Function MyFunction(Argument) MyFunction = (Argument + 5) * 10 End Function</pre>
Subroutine	<pre>Sub UsingFunction() MyNumber = InputBox("Please enter a number", Title:="MyFunction") msgbox "The number is " &amp; MyFunction(MyNumber), Title:="MyFunction" End Sub Sub UsingFunction() MyNumber = InputBox("Please enter a number", Title:="MyFunction") msgbox "The number is " &amp; MyFunction(MyNumber), Title:="MyFunction" End Sub</pre>
Input Window	
Output	

12. [1 point] Explain how argument passing differs between a function that returns a value and a function that does not return a value.

**A:** For a function that returns a value, the passing argument must be enclosed in parenthesis. For a function that does not return a value, the passing arguments are listed without parenthesis. By assigning a value to the name of the function itself VBA knows to send the assigned back as the return value of the function. Otherwise, if we do not take this step, the function will not return a value.

13. [1 point] With regards to the "msgbox" function, give an example of named vs. unnamed arguments.

**A:** Function MsgBox has the following arguments prompt[, buttons] [, title] [, helpfile, context]

---

**MsgBox "Hello",,"This is a Title" is an example where unnamed arguments are used.**

**MsgBox Prompt:="Hello", Title:="This is a title" is an example where the arguments prompt and title are named as "Prompt:=" and "Title:=".**

**Naming arguments is useful for calling procedures that have a large number of arguments and it is mostly optional.**

14. **[1 point]** *What is the difference in scope for a variable declared outside of a subroutine and a variable declared inside a subroutine?*

**A: A variable or constant declared inside a subroutine is not visible outside that subroutine. A variable or constant declared outside subroutine is visible to all other subroutines.**

15. **[1 point]** *What is the difference in variable scope when using the keyword "public" and "private" outside of a subroutine?*

**A: Public variables are available to all procedures in all subroutines in a project, private variables are available only to procedures in that subroutine. By default, variables declared with the Dim statement in the Declarations section are scoped as private.**

16. **[1 point]** *What is the difference between a sub procedure and a function procedure?*

**A: There are two types of procedures in VBA: *Function procedures* and *sub procedures* (i.e. *subroutines*). A function is a procedure that performs an operation and can return a value. A sub procedure also performs an operation, but it doesn't return a value.**

17. **[1 point]** *What is the difference between a property, a method and an event with regards to an object. Provide a concrete example of each.*

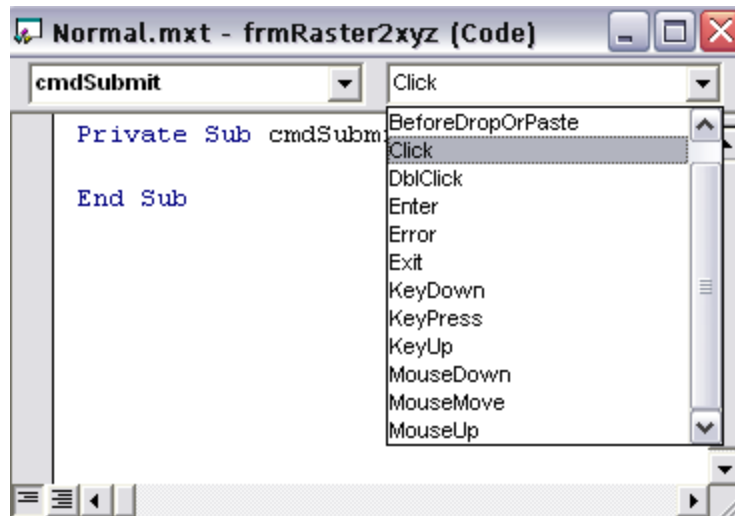
**A: A property is an attribute of an object that defines one of the object's characteristics, such as size, color, or screen location, or an aspect of its behavior, such as whether it is enabled or visible. In the example below, the Caption property is changed.**

```
Sub ChangeName(newTitle)
    myForm.Caption = newTitle
End Sub
```

**A method is an action that an object can perform. The following procedure uses the Add method to add a new item to a ComboBox**

```
Sub AddEntry(newEntry as String)
    Combol.Add newEntry
End Sub
```

An event is an action recognized by an object, such as clicking the mouse or pressing a key, and for which one can write code to respond. Events can occur as a result of a user action or program code, or they can be triggered by the system.



An list of events that can be recognized by an command button

18. [1 point] On a UIControlButton, why is the click event important?

**A:** Click is in this case event handler( an event is always associated with an object). It simply means that when the user *Click* on the UIButtonCONTROL, the code will be executed otherwise there will be no code execution. Visual basic is event-driven, the user dictates program flow and code executes only when events have occurred.