

Extra VBA-review questions: the following questions are meant to provide you with some extra practice so you need to actually try them on your own to get anything out of it. For that reason, solutions won't be posted and I won't just "email you the solutions". But it really shouldn't be a big loss because I have taught you how you can find out the answers for yourself:

- For the 'tracing' questions where you have to determine the output of a program or the result of an expression then you can always type in the program or expression and run it yourself.
- The 'writing' questions (e.g. write a program) are a little trickier but you can still check your answers by testing the results e.g. does your program or query produce the results specified in the questions.

If you are still unsure of things after you have attempted a question (e.g. you can't figure out why you got a given result after typing it in and running the program), then you can ask (just be sure to show me the work that you have done so I can see how far that you have gotten and perhaps where you are having problems). Again don't just coming in cold without making an attempt. I'm happy to help, however you should make sure you provide a reasonable lead time if you have questions. In extreme cases (e.g. getting emailed questions at all hours of the night before an 8:30 AM exam) you probably won't get a response until **after** the exam.

CPSC 230 Extra review and solutions

Multiple choice questions:

Unless otherwise specified you can assume that the programs are correct and do not contain any syntax errors (e.g. the 'sub' has been defined, all variables have been declared via a 'Dim' etc.)

For Questions 1 – 6 determine the output of the MsgBox

1)

```
x = 12
s = ""
If (x > 0) Then
    s = s & "a"
End If
s = s & "b"
MsgBox (s)
```

- a
- b
- s
- ab
- None of the above

2)

```
x = -12
s = ""
If (x > 0) Then
    s = s & "a"
Else
    s = s & "b"
End If
s = s & "c"
MsgBox (s)
```

- a. c
- b. s
- c. ac
- d. bc
- e. None of the above

3)

```
x = 11
s = ""
If (x > 0) Then
    s = s & "a"
End If
If (x > 10) Then
    s = s & "b"
End If
If (x > 100) Then
    s = s & "c"
End If
MsgBox (s)
```

- a. a
- b. b
- c. c
- d. ab
- e. abc

4)

```
x = 11
s = ""
If (x > 0) Then
    s = s & "a"
ElseIf (x > 10) Then
    s = s & "b"
ElseIf (x > 100) Then
    s = s & "c"
End If
MsgBox (s)
```

- a. a
- b. b
- c. c
- d. ab
- e. abc

5)

```
x = -1
y = 1
s = ""
If ((x > 0) And (y > 0)) Then
    s = s & "a"
Else
    s = s & "b"
End If
s = s & "c"
MsgBox (s)
```

- a. a
- b. b
- c. ac
- d. bc
- e. None of the above

6)

```
x = -1
y = 1
s = ""
If ((x > 0) Or (y > 0)) Then
    s = s & "a"
Else
    s = s & "b"
End If
s = s & "c"
MsgBox (s)
```

- a. a
- b. b
- c. ac
- d. bc
- e. None of the above

For Questions 7 – 13 specify the number of times that the loop will execute.

7)

```
i = 1
Do While (i < 4)
    MsgBox (i)
    i = i + 1
Loop
```

- a. 0
- b. 1
- c. 3
- d. 4
- e. Loop never ends

8)

```
i = 0
Do While (i < 4)
    MsgBox (i)
    i = i + 1
Loop
```

- a. 0
- b. 1
- c. 3
- d. 4
- e. Loop never ends

9)

```
i = 0
Do While (i <= 4)
    MsgBox (i)
    i = i + 1
Loop
```

- a. 0
- b. 3
- c. 4
- d. 5
- e. Loop never ends

10)

```
i = 0
Do While (i < 4)
    MsgBox (i)
Loop
i = i + 1
```

- a. 0
- b. 3
- c. 4
- d. 5
- e. Loop never ends

11)

```
i = 1
Do While (i < 21)
    MsgBox (i)
    i = i * 3
Loop
```

- a. 0
- b. 1
- c. 3
- d. 20
- e. 21

12)

```
i = 10
Do While (i > 4)
    MsgBox (i)
    i = i - 2
Loop
```

- a. 0
- b. 2
- c. 3
- d. 4
- e. Loop never ends

13)

```

i = 4
Do While (i < 4)
    MsgBox (i)
    i = i / 2
Loop

```

- a. 0
- b. 2
- c. 3
- d. 4
- e. Loop never ends

For Questions 14 – 17 determine the output of the MsgBox

14) (May be counted as 'very hard' or beyond your exam difficulty if nested loops weren't covered during the term)

```

k = 0
i = 1
Do While (i <= 4)
    j = 1
    Do While (j <= 3)
        j = j + 1
        k = k + 1
    Loop
    i = i + 1
Loop
MsgBox (i & " " & j & " " & k)

```

- a. 4 3 7
- b. 4 3 12
- c. 5 4 12
- d. 5 4 20
- e. None of the above

15)

```
s = ""
x = 66
y = -66
z = 0
If (x > 12) Then
    s = s + "a"
    If (y < 0) Then
        s = s + "b"
    End If
    If (z > 0) Then
        s = s + "c"
    End If
End If
s = s + "d"
MsgBox (s)
```

- a. a
- b. ad
- c. abd
- d. acd
- e. abcd

16)

```
s = ""
x = 66
y = -66
Z = 0
If (x > 12) Then
    s = s + "a"
    If (y < 0) Then
        s = s + "b"
        If (Z > 0) Then
            s = s + "c"
        End If
    End If
End If
s = s + "d"
MsgBox (s)
```

- a. a
- b. ad
- c. abd
- d. acd
- e. abcd

17)

```
i = 0
sum = 0
Do While (i < 6)
  If (i <= 3) Then
    sum = sum + i
  End If
  i = i + 1
Loop
MsgBox (sum)
```

- a. 6
- b. 9
- c. 21
- d. 123456
- e. None of the above

18) What's the output of the MsgBox when the user enters 1, 2, 3, 4, 5, 6, -1 as inputs for the InputBox?

```
temp = 1
sum = 0
Do While (temp > 0)
  temp = InputBox("Enter a num: ")
  If ((temp Mod 2) = 0) Then
    sum = sum + temp
  End If
Loop
MsgBox (sum)
```

- a. 9
- b. 12
- c. 21
- d. 123456
- e. None of the above

- 19) This VBA program has been written within Excel (it's contained with an .xlsm file). Given the Excel spreadsheet data shown below, after running the VBA program what will be the output appear in the MsgBox?

	A	B	C
1	1	2	1
2	3	2	1
3	4	5	1

```
x = ""
If (Range("A1").Value > Range("C2").Value) Then
    x = Range("B3").Value
Else
    x = Range("C1").Value
End If
MsgBox (x)
```

- 1
 - 2
 - 5
 - B3
 - C1
- 20) This **VBA program has been written within PowerPoint** (it's contained with a .pptm file). What is the result of running the following program?

```
ActivePresentation.Slides.Add Index:=1, Layout:=ppLayoutBlank
```

- A new slide has been inserted at the beginning
- A new slide has been inserted at the end
- 'x' number of new slides have been inserted at the end, even numbered slides are blank, odd numbered ones have a title
- (a) & (b)
- The number of slides that have a blank layout are counted.

21) This **VBA program has been written within PowerPoint** (it's contained with a .pptm file). What is the result of running the following program?

```
x = ActivePresentation.Slides.Count + 1
ActivePresentation.Slides.Add Index:=x, Layout:=ppLayoutBlank
```

- A new slide has been inserted at the beginning
- A new slide has been inserted at the end
- 'x' number of new slides have been inserted at the end, even numbered slides are blank, odd numbered ones have a title
- (a) & (b)
- The number of slides that have a blank layout are counted.

22) What will be the appearance (in a web browser) of a document containing the following html tags?

```
ab<b>cdef<br>g</b>hij
```

- abc**def**
ghij
- abc**def**
ghij
- abc**def**
ghij
- abc**def**
ghij
- abcdef
ghij

23) What will be the appearance (in a web browser) of a document containing the following html tags? (This is an example of complex although not necessary hard – if you really understand html tags work).

```
<ol>
<li>aaa</li>
<li>bb</li>
</ol>
<ul>
<li>ccc</li>
</ul>
dd
```

a.
aaa
bb

ccc

dd

1. aaa
2. bb

• ccc
dd

b.

• aaa
• bb

1. ccc
dd

c.

1. aaa
2. bb

1. ccc
dd

d.

• aaa
• bb

• ccc
dd

e.

Short answer:**Short answer 1**

Given the following inputs, what's the output of the MsgBox?

Input: 0, 0, 0 Write your output here:

Input: 0, 1, 101: Write your output here:

Try predicting the output with other inputs, here's some examples (try additional ones)

Input: -1, -1, -1

Input: 1, 1, 11

Input: 2, -2, 1000

Input: 10, 100, 1000

```
Sub sa1()
    Dim num1 As Long
    Dim num2 As Long
    Dim num3 As Long
    Dim string1 As String

    num1 = -1
    num2 = -1
    num3 = -1
    string1 = ""

    num1 = InputBox("Enter a number", "")
    num2 = InputBox("Enter a number", "")
    num3 = InputBox("Enter a number", "")

    If ((num1 > 0) And (num2 > 0)) Then
        string1 = "a"
    End If
    If (num3 > 10) Then
        string1 = string1 + "A"
    End If
    If ((num1 > 0) Or (num2 >= 0)) Then
        string1 = string1 + "b"
        If (num3 > 100) Then
            string1 = string1 + "c"
        End If
    End If
    MsgBox (string1)
End Sub
```

Short answer 2:

Modify the following VBA program so it will display “Match” if age is 18 – 25 and city is either “Calgary” or “Red Deer”. “Not a match” should be displayed in all other situations.

Output messages are to be displayed via MsgBox popups.

```
Sub branchProblem()  
    Dim age As Long  
    Dim city As String  
  
    age = InputBox("Age (e.g. 18): ")  
    city = InputBox("City (e.g. Edmonton): ")  
    ' Write your answer here
```

```
    ' End of answer space  
End Sub
```

Short answer 3:

Modify the following VBA program so it will repeatedly prompt the user for a password (which should be stored in the variable 'userEnteredPassword') until the user enters the correct password (which is the "SYSTEM_PASSWORD"). Also each time that the passwords don't match the program should display an appropriate error message. When the passwords do match the program will stop prompting and instead display a message "Login successful"

```
Sub errorHandlingProgram()  
    Const SYSTEM_PASSWORD = "password"  
    Dim userEnteredPassword As String
```

```
    ' Write your answer here
```

```
    ' End of answer space  
End Sub
```

Short answer 4 (an example of a ‘hard’ final examination question)

Modify the following VBA program so the variable ‘result’ will be the resulting exponent of ‘base’ raised to the value stored in the variable ‘power’:

- Base = 2, power = 3, result = $2^3 = 8$
- Base = 1, power = 12, result = $1^{12} = 1$
- Base = 55, power = 0, result = $55^0 = 1$

You must not use functions or methods built into VBA that will calculate an exponent for you. Instead you must write the code yourself. Recall: that an exponent is a series of successive multiplications ($2^3 = 2 * 2 * 2$) so some sort of looping mechanism needs to be employed. For this version of the question you can assume that the user will enter a base and power that is zero or greater.

JT’s hint: If this were an actual exam question even if you can’t figure out how to calculate an exponent try to get partial marks and write the parts that you can visualize.

```
Sub exponentLoops()
    Dim base As Long
    Dim power As Long
    Dim result As Long
    Dim i As Long

    base = InputBox("Base (zero or greater): ")
    power = InputBox("Exponent (zero or greater): ")

    ' Write your answer here

    ' End of answer space
    MsgBox (base & " raised to " & power & "=" & result)
End Sub
```

Short answer 5: example of complex trace (hard)

Specify the output text shown in the MsgBox when the following VBA program is run.

```

Sub extraTrace()
    Dim i As Long
    Dim result As Long

    i = 1
    result = 0
    Do While (i <= 20)
        If (i < 5) Then

            i = i + 1
        ElseIf (i <= 10) Then

            i = i + 2
        Else

            i = i * 2
        End If
        result = result + 1
    Loop
    MsgBox ("i=" & i & " result=" & result)
End Sub

```

' Write your answer here

Hint: Inserting a MsgBox into each of the above 3 IF-branches can make it easier to determine how this result was derived. Go a step and at time, write down intermediate results, don't try to do it all "in your head",

'i' from range 1 to 4 (first 'if' runs 4 times)

* Result increased to 4

When I = 5 then first case no longer applies, second case applies

'i' from range 5 – 10 (i = 5, 7, 9, 11: second 'if' runs 4 times)

* Result increased to 11

When I = 11 then second case doesn't apply, third case applies

'i' from range 11 – 20 (i = 11: last 'if' runs 1 time)

When 'I' doubles from 11 to 22, 'result' no longer changes because the Boolean for the loop is false and the loop stops executing

Short answer 6: example of complex trace (hard). Hint: Go slow and write down intermediate results of your trace. Don't do it in your head.

```

Sub practiceTrace()
  Dim x As Long
  Dim y As Long
  Dim z As Long
  Dim a As String
  Dim b As String
  Dim c As String
  x = 2
  y = 3
  z = -1
  c = ""
  a = "!"
  b = "#"
  If (True) Then
    c = c & "1"
    a = "aa"
    b = "aa"
    If ((y - x) <= z) Then
      c = c & "2"
      If (x > 0) Then
        c = c & "3"
      End If
      If (x > 1) Then
        c = c & "4"
      End If
    Else
      c = c & "5"
      z = 1
      Do While (z > x)
        c = c & z
        z = z - 1
      Loop
    End If
  Else
    c = c & "z"
    If ((a = b) And (x < y)) Then
      c = c & "c"
    Else
      c = "icu"
    End If
  End If
  MsgBox (c)
End Sub

```

Short answer 7: code writing

The hottest dating site on net is “Swipe right: Plenty O’ Tam” ©. It uses the proprietary match-making secret algorithm that generates a ‘Tamdex’™ score. The result of the score determines the compatibility level between you and another member. Augment the program below so that the appropriate popup message appears for all the possible Tamdex scores shown in the table.

Tamdex score	Match level
0 – 349	Better swipe left now!
350 - 499	Yawn...
500 - 699	Excellent match
700 - 799	Cupid recommended: you’ll laugh, you’ll cry, that person will become a part of you.
800 - 888	Perfect match

```
Sub dating()
```

```
    Dim tDex As Long
```

```
    ' You don't need know how the instruction below works
```

```
    ' (Plenty of Tam algorithm is TM and secret!) Just assume
```

```
    ' that the tDex will contain a value between 1 - 888.
```

```
    tDex = Int(888 * Rnd) + 1
```

```
    ' Write your answer here
```

```
    ' End of answer space
```

```
End Sub
```