COMP1022Q
Introduction to Computing with Excel VBA

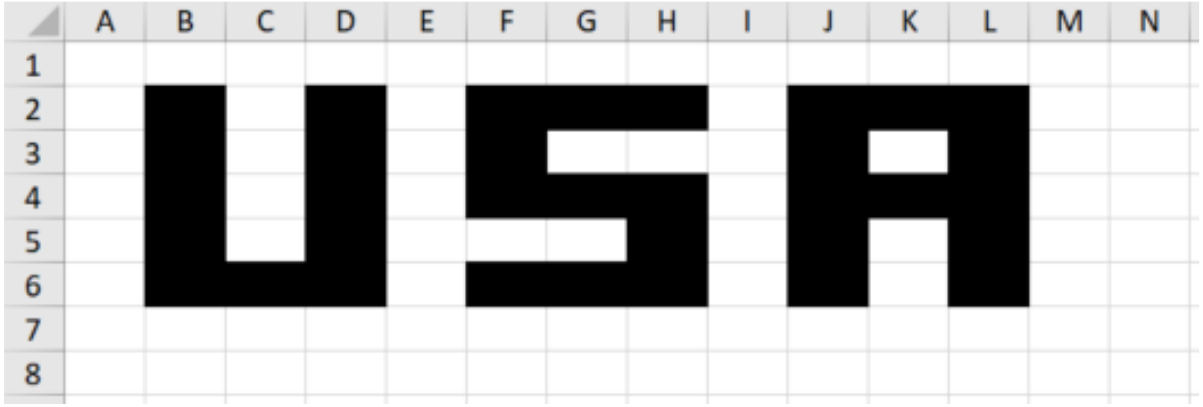
Final Week
Lecture Review
Questions and Answers

David Rossiter, Gibson Lam and Cecia Chan

```
Sub Workbook_Open()
    Dim Row As Integer, Col As Integer

    For Row = 2 To 6
        For Col = 2 To 14

            If (Col = 2 Or Col = 4 Or (Col >= 2 And Col <= 4 And Row = 6)) Or _
                (Col >= 6 And Col <= 8 And (Row = 2 Or Row = 4 Or Row = 6)) Or _
                (Row = 3 And Col = 6) Or _
                (Row = 5 And Col = 8) Or _
                (Col = 10 Or Col = 12) Or _
                ((Col >= 10 And Col <= 12) And (Row = 2 Or Row = 4)) Then

                Cells(Row, Col).Interior.ColorIndex = 1
            End If

        Next Col
    Next Row
End Sub
```

Q1) What is the result?

colour 1 is black



Q2) Here is some VBA code.

```
Sub Button1_Click()
    Sum = 0
    Index = 0
    Do While Index < 50
        Sum = Sum + Cells(Index + 5, 1).Value
        Index = Index + 1
    Loop
    MsgBox Sum / Index
End Sub
```

5 to 54 A

You need to replace the content of `Button1_Click` with **a single line of VBA code** that gives an equivalent result.

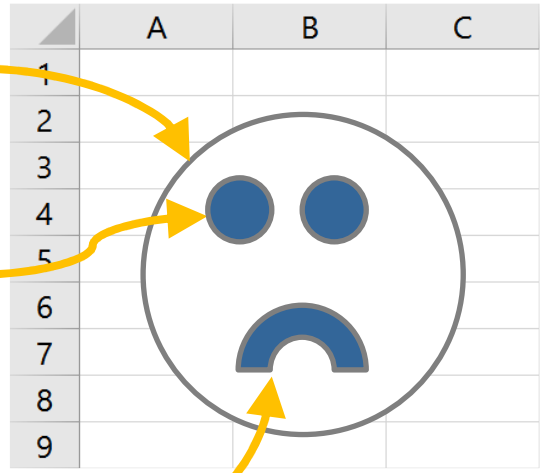
Complete the line which begins with `MsgBox` shown below.

```
Sub Button1_Click()

    MsgBox WorksheetFunction.Average(Range("A5:A54"))
End Sub
```

Q3) What does the following code produce?

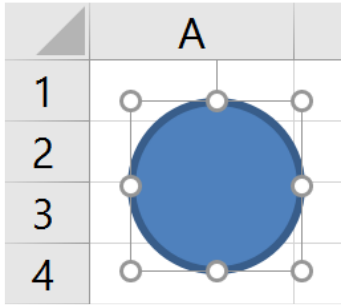
```
Dim Shape1 As Shape
Set Shape1 = ActiveSheet.Shapes.AddShape( _
             msoShapeOval, 20, 20, 100, 100)
Shape1.Fill.ForeColor.RGB = vbWhite

ActiveSheet.Shapes.AddShape msoShapeOval, _
                            40, 40, 20, 20
ActiveSheet.Shapes.AddShape msoShapeOval, _
                            70, 40, 20, 20

Dim Shape2 As Shape
Set Shape2 = ActiveSheet.Shapes.AddShape( _
                 msoShapeBlockArc, 50, 80, 40, 40)
```

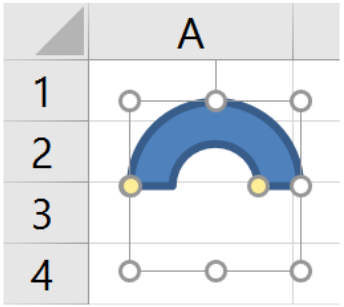
Answer:



The following information is very useful for this question.

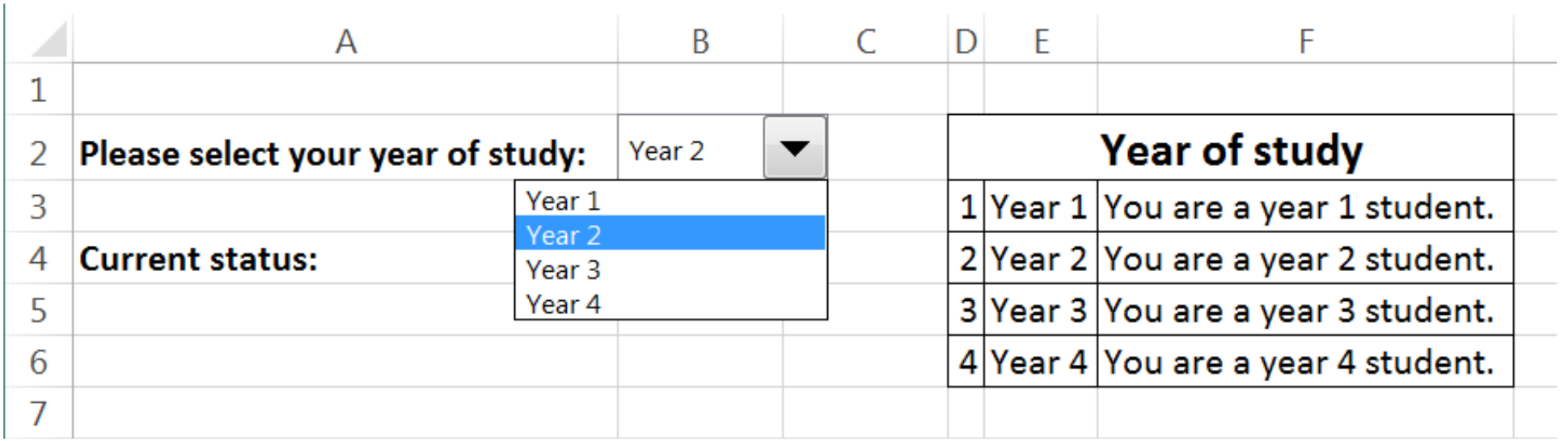
```
ActiveSheet.Shapes.AddShape _
    msoShapeOval, 10, 10, 40, 40
```





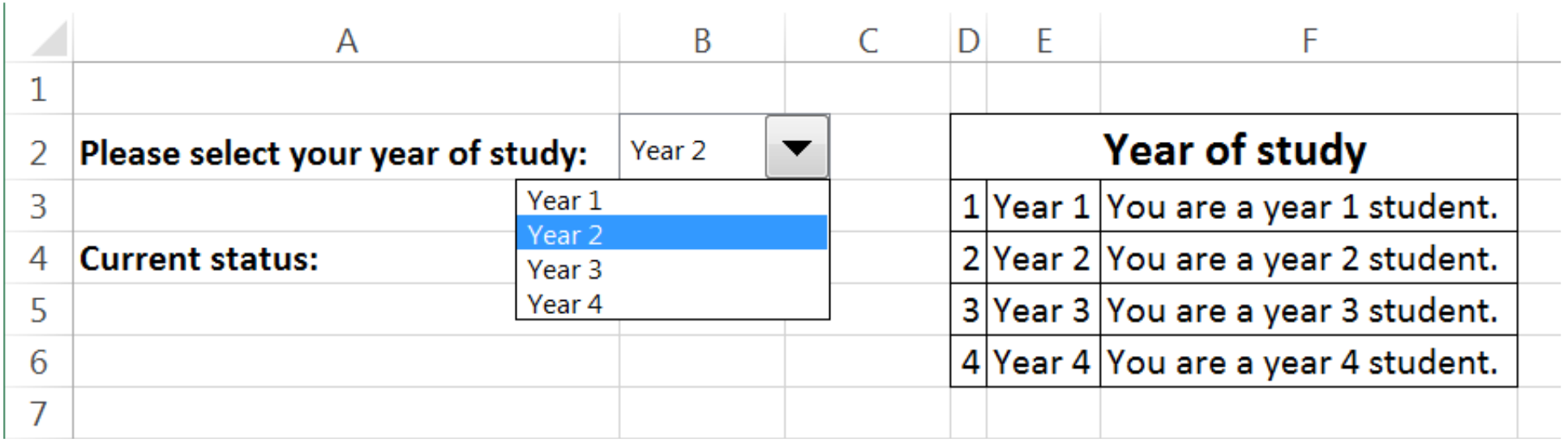

```
ActiveSheet.Shapes.AddShape _
    msoShapeBlockArc, 10, 10, 40, 40
```

Q4) Here is a worksheet with a Form combo box control.



The *Input range* of the combo box control is set to $**E$3:$E$6** and the *Cell link* is set to Cell **B2** (hidden under the combo box control).

There is **no** VBA code associated with the workbook, the worksheet or the Form control.



Which one of the following formulas can be used in cell **A5** if you want to show, for example, "You are a year 2 student." in the cell after selecting "Year 2" in the combo box control?

A) =VLOOKUP(B2, $E$3:$F$6, 2)

B) =VLOOKUP(B2, $E$3:$F$6, 2, FALSE)

C) =VLOOKUP(B2, $D$3:$F$6, 3)

D) =HLOOKUP(B2, $D$3:$F$6, 3)

E) =VLOOKUP(B2, $D$3:$D$6, $F$3:$F$6)

Q5A) Grade points are numeric representations of letter grades. The conversion of letter grades to grade points is shown below.

Letter Grade	A+	A	A-	B+	B	B-	C+	C	C-	D	F
Grade Point	4.3	4	3.7	3.3	3	2.7	2.3	2	1.7	1	0

Given a letter grade, the VBA function shown below returns the corresponding grade point. Please fill in the blank in the code so that it works correctly.

```
Function GradePoint(ByVal Grade As String) As Double
  For Pos = 1 To 5
    If Left(Grade, 1) = Mid( "FDCBA" , Pos, 1) Then
      GradePoint = Pos - 1
    End If
  Next
  If Right(Grade, 1) = "+" Then
    GradePoint = GradePoint + 0.3
  ElseIf Right(Grade, 1) = "-" Then
    GradePoint = GradePoint - 0.3
  End If
End Function
```

The position of the letters “F” to “A” is at 1 to 5 respectively

Q5B) The VBA code shown on the next slide calculates the CGA of the courses taken by a student. The courses, their number of credits and letter grades are shown in the table below:

	B	C	D
2	**Course code**	**Number of credits**	**Letter grade**
3	COMP1022Q	3	B+
4	ENGG1130	3	B-
5	CIVL1100	3	A-
6	CENG1500	3	C+
7	MATH1020	4	A+
8	SOSC1960	3	D
9	MECH1906	3	A
10	ELEC1100	4	B
11	LANG1002	3	C-
12	LANG1003	2	F
13			
14	Your CGA is	2.752	

The CGA of the courses is calculated by a weighted average of the grade points. The weight of each grade point is the corresponding credit of the course.

You need to fill in the blanks in the code so that the code calculates the CGA correctly.

The following code uses the `GradePoint` function.

	B	C	D
2	**Course code**	**Number of credits**	**Letter grade**
3	COMP1022Q	3	B+
4	ENGG1130	3	B-
5	CIVL1100	3	A-
6	CENG1500	3	C+
7	MATH1020	4	A+
8	SOSC1960	3	D
9	MECH1906	3	A
10	ELEC1100	4	B
11	LANG1002	3	C-
12	LANG1003	2	F
13			
14	Your CGA is	2.752	

```
Dim CGP As Double,
    TotalCGP As Double
Dim Row As Integer,
    TotalCredits As Integer
TotalCGP = 0
TotalCredits = 0
For Row = 3 To 12
  CGP = Cells(Row, 3).Value * _
        GradePoint( Cells(Row, 4).Value )
  TotalCredits = TotalCredits + Cells(Row, 3).Value
  TotalCGP = TotalCGP + CGP
Next
Cells(14, 3).Value = TotalCGP / TotalCredits
```

Q6) The following VBA code highlights some rows using green. Which rows?

	A	B		C	
1	**Name**	**Height**		**Weight**	
2	Donelle	1	180	6	70
3	Davinia	4	174	8	56
4	Alise	6	153	2	78
5	Judith	10	133	7	62
6	Sissie	6	153	10	54
7	Tracee	2	177	9	55
8	Aliah	8	139	5	72
9	Agatha	9	135	1	79
10	Gussie	3	176	3	75
11	Harley	5	160	4	73

```
Dim Row As Integer,
    Column As Integer

For Row = 2 To 11
  For Column = 1 To 3
    If WorksheetFunction.Rank( _
          Cells(Row, 2).Value, _
          Range("B2:B11")) >= 8 _
       Or _
       WorksheetFunction.Rank( _
          Cells(Row, 3).Value, _
          Range("C2:C11")) <= 3 Then
      Cells(Row, Column).Interior.Color = vbGreen
    End If
  Next
Next
```

Q7) Here is the the VBA code that modifies some selected cells

```
Dim Cell As Range
For Each Cell In Selection
  If Cell.Value = "student" Then
    Cells(Cell.Row, Selection.Column + _
        Selection.Columns.Count - 1 ).Value = "discount"
  End If
Next Cell
```

Suppose that you select some cells, as shown below:

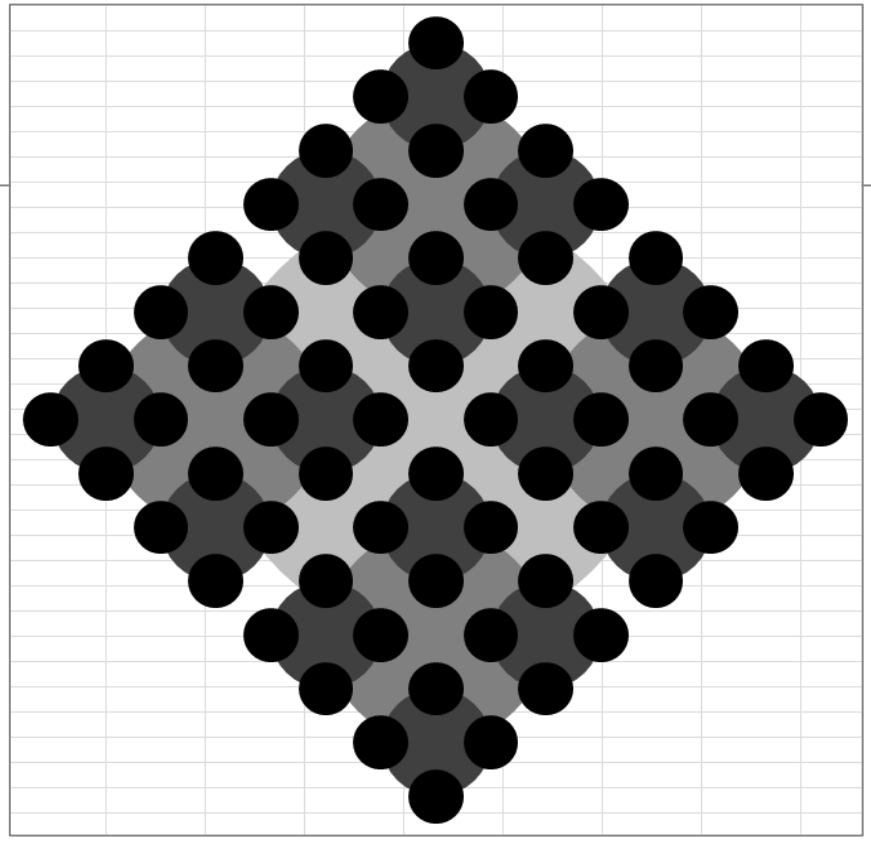
	A	B	C	D	E
1					
2	Adam	24	clerk		
3	Sharon		student	discount	
4	Cathy	67	teacher		
5	Denny	19	student	discount	
6	Agnes	11	student	discount	
7	Kevin	33	Technician		
8	Christine	18	Sales		
9					

Then you run the VBA code, what will happen?

Q8) Here is some VBA code which uses a recursive subroutine shown on the next slide.

```
Dim MaxDepth As Integer

Sub InputMaxDepth()
    MaxDepth = InputBox("Enter the maximum depth " & _
                        "for drawing the circles: ")
    DrawCircle 250, 350, _
               128, 0
End Sub
```

What is the number you need to enter into the input box so that the image on the right is created?



The recursive subroutine is shown below:

```
Sub DrawCircle(ByVal X As Double, ByVal Y As Double, _
               ByVal Radius As Double, ByVal Depth As Integer)
    Dim Gray As Double, Size As Double
    Dim CircleObj As Shape
    If MaxDepth > 0 Then
        Gray = 255 * (CDbl(MaxDepth - Depth) / (MaxDepth + 1))
    End If
    Size = Radius * 2

    Set CircleObj = ActiveSheet.Shapes.AddShape( _
        msoShapeOval, X - Radius, _
        Y - Radius, Size, Size)
    CircleObj.Line.Visible = False
    CircleObj.Fill.ForeColor.RGB = _
        RGB(Gray, Gray, Gray)

    If Depth < MaxDepth Then
        DrawCircle X + Radius, Y, _
                   Radius / 2, Depth + 1
        DrawCircle X - Radius, Y, _
                   Radius / 2, Depth + 1
        DrawCircle X, Y - Radius, _
                   Radius / 2, Depth + 1
        DrawCircle X, Y + Radius, _
                   Radius / 2, Depth + 1
    End If
End Sub
```

Answer: 3