COMP1022Q
Introduction to Computing with Excel VBA

Drawing Excel Shapes

David Rossiter and Gibson Lam

Outcomes

- After completing this presentation, you are expected to be able to:
  1. Create Excel shapes in VBA
  2. Change the appearance of shapes using VBA
  3. Create nested loops which use shapes

Remember a *nested loop* simply means a loop (any kind of loop) inside a loop

Adding a Shape Using VBA

- Some of our examples will use shapes
- This is the way to add a shape using VBA

This is a number, or a variable containing a number, which indicates the shape you want to draw

(1 is a square, 2 is a rhombus, and so on)

```
ActiveSheet.Shapes.AddShape ShapeNumber, _
                            X, Y, Width, Height
```

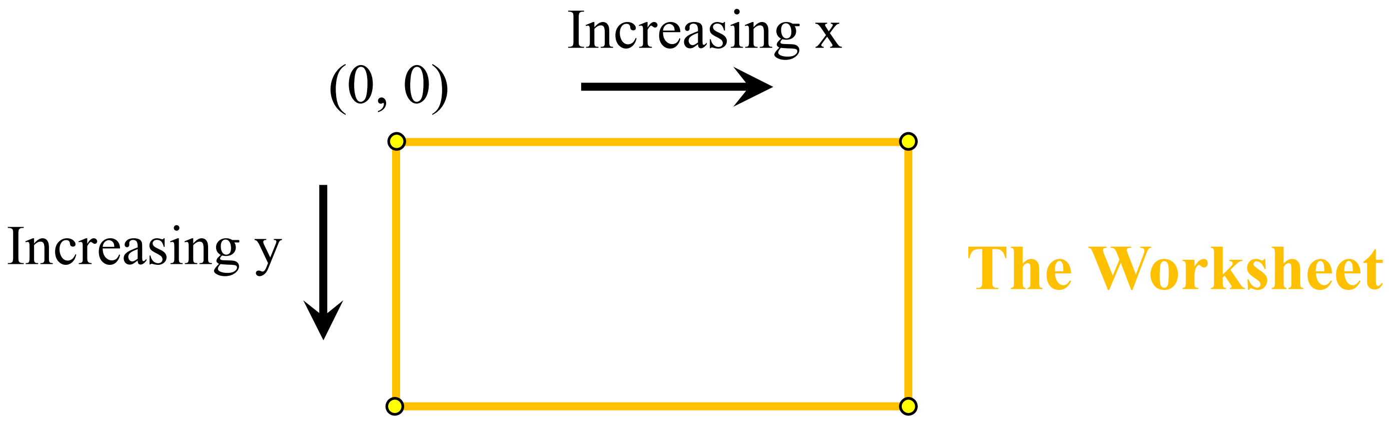
This is the currently selected worksheet

`X` *and* `Y` *is the top-left corner of the shape*

`Width` *and* `Height` *specify the size of the shape*

Coordinate System in Excel

- Shapes are added to the worksheet, but they don't go in cells
- Shapes are separate from cells
- For shapes, (0, 0) is the top left corner of the worksheet
  - Increasing values of x go across the page, to the right
  - Increasing values of y go down the page



Drawing Different Shapes

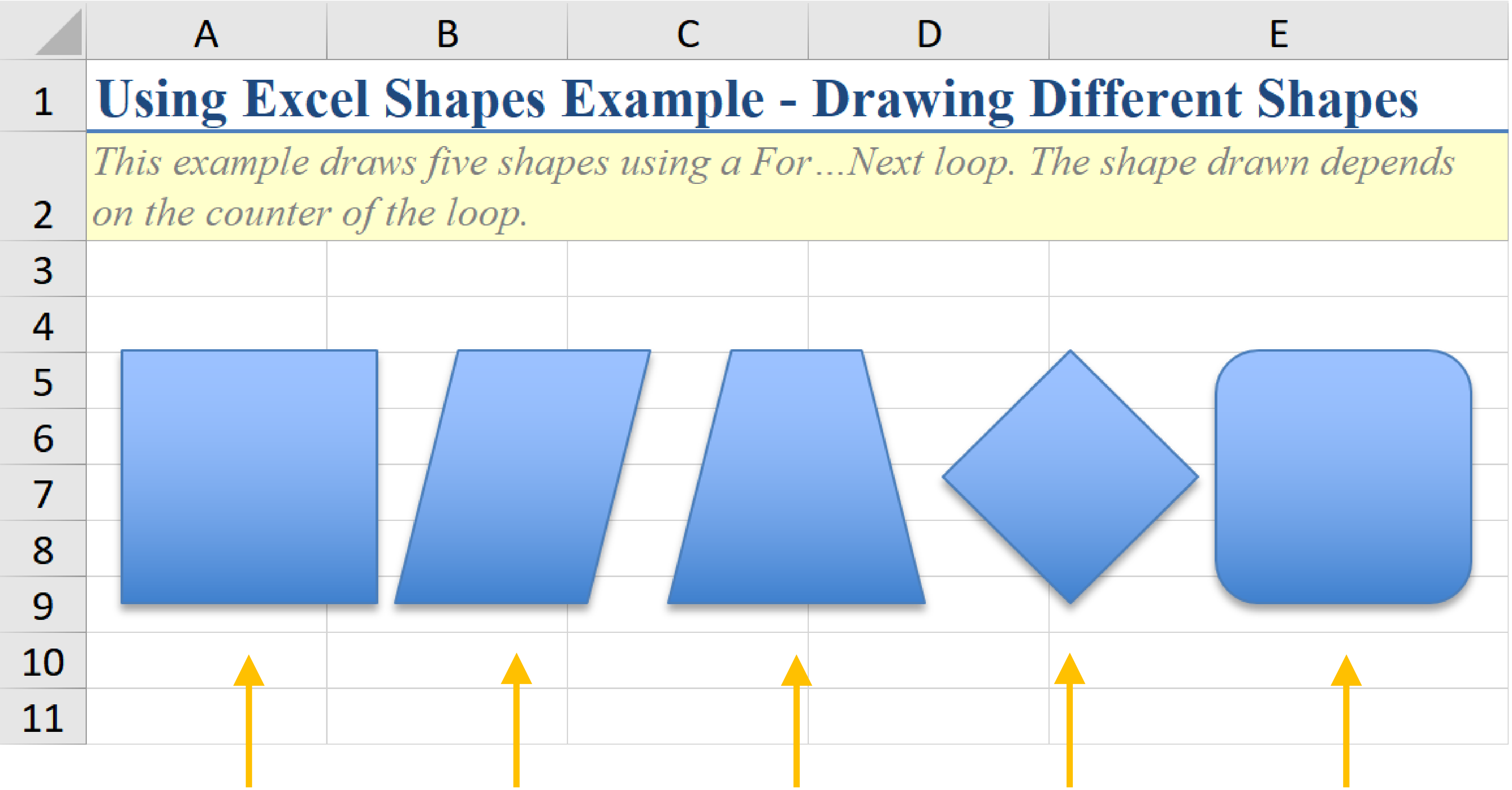
```
Dim ShapeNumber As Integer
Dim X As Integer
X = 10

For ShapeNumber = 1 To 5
    ' Draw a shape on the worksheet
    ActiveSheet.Shapes.AddShape ShapeNumber,_
        X, 80, 70, 70

    ' Increase X position for the next shape
    X = X + 75
Next ShapeNumber
```

The loop variable

Add a shape at (X, 80) with a size of 70 by 70 and the shape number specified by the control variable

x, y, width, height

The Result



Using Excel Shapes Example - Drawing Different Shapes

This example draws five shapes using a For…Next loop. The shape drawn depends on the counter of the loop.

Shape = 1 *Shape = 2* *Shape = 3* *Shape = 4* *Shape = 5*

Shape Names

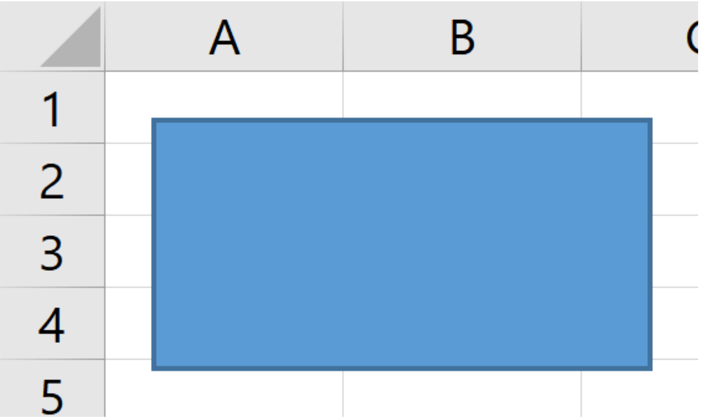
- In the previous example, we used numbers to refer to the shapes we wanted to create
- Sometimes the code looks better if we use the *names* of the shapes instead of some numbers when we create the shapes, e.g.:
  - `msoShapeRectangle` (= shape number 1)
  - `msoShapeOval` (= shape number 9)
- You can find more shape names here:

https://msdn.microsoft.com/en-us/library/office/ff862770.aspx

Making a Rectangle

- As an example, we can create a rectangle using the name `msoShapeRectangle`, instead of 1:

```
Set Rect = ActiveSheet.Shapes.AddShape( _
        msoShapeRectangle, 10, 10, 100, 50)
```



- In the code above we put the newly created rectangle shape in a variable `Rect`
- This is because we can then change the appearance of the rectangle using the variable, if we want to

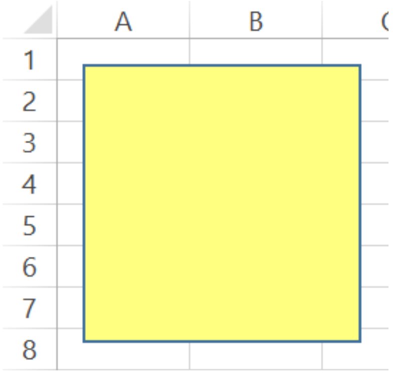
Changing the Appearance of Shapes

- To change the appearance of Excel shapes, you need to do that using some shape properties
- We will look at how to do these:
  - Changing the fill colour of shapes
  - Changing the outline of shapes
  - Rotating a shape

- By default, an Excel shape has a light blue fill colour and a thin blue outline

Changing the Fill Colour of Shapes

- The fill colour of a shape is changed using the `Fill.ForeColor.RGB` property
- For example, if a square has been created and stored in a variable `Square`, the following code can change its colour to light yellow:



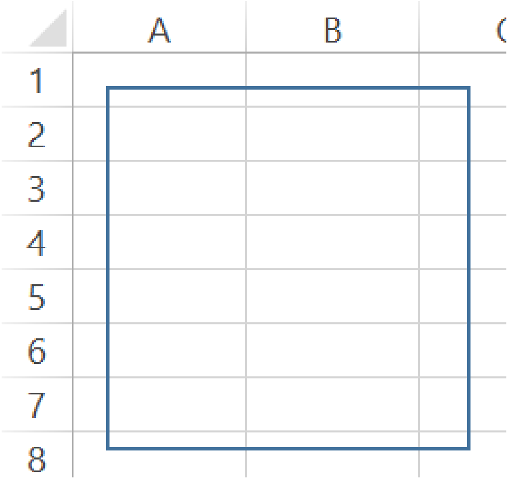
```
Square.Fill.ForeColor.RGB = vbYellow
```

- You cannot use ColorIndex for Excel shapes
- If we have time we will look at what exactly RGB means later in the course

Making Hollow Shapes

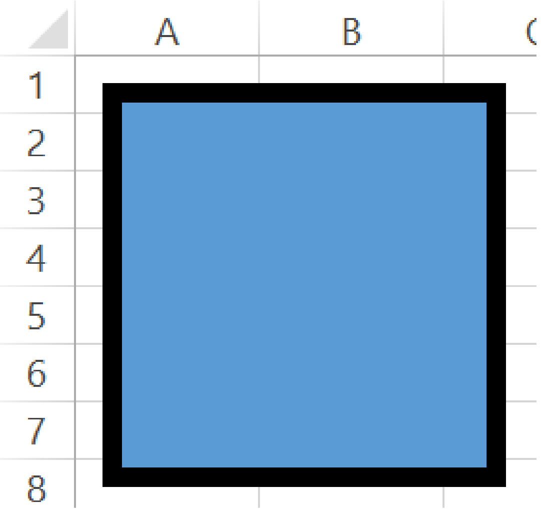
- Since ColorIndex is not available to Excel shapes, you cannot ‘clear’ the shape colour by setting ColorIndex to 0, which you can do for a cell background
- To make hollow shapes you need to tell Excel to make the fill colour ‘invisible’ using this code:

```
Square.Fill.Visible = False
```



Changing the Outline of Shapes

- The shape outline colour is controlled by the `Line.ForeColor.RGB` property
- Similar to the fill colour, you cannot use ColorIndex
- To change the thickness of the outline, a number can be assigned to the `Line.Weight` property
- For example, the following code makes a thick black outline for a square shape:

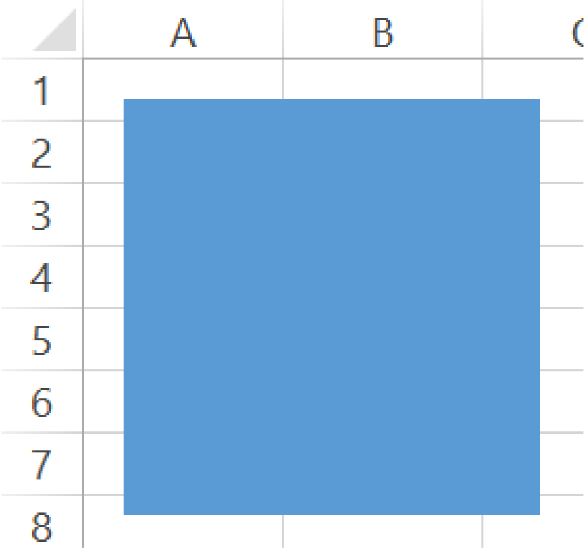


```
Square.Line.ForeColor.RGB = vbBlack
Square.Line.Weight = 5
```

Removing the Shape Outline

- To remove the shape outline you can make it invisible using this code:

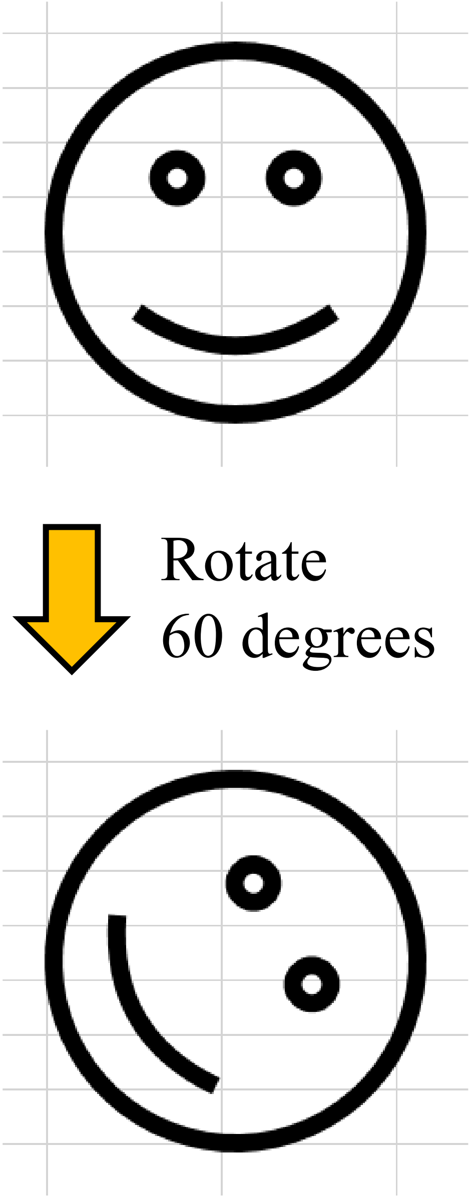
```
Square.Line.Visible = False
```



Rotating a Shape

- The `Rotation` property of a shape changes the orientation of the shape
- The value of this property is an angle between 0 to 360
- The shape is rotated in clockwise direction based on this value
- For example, you can rotate a face shape (in a variable `Face`) 60 degrees using this code:

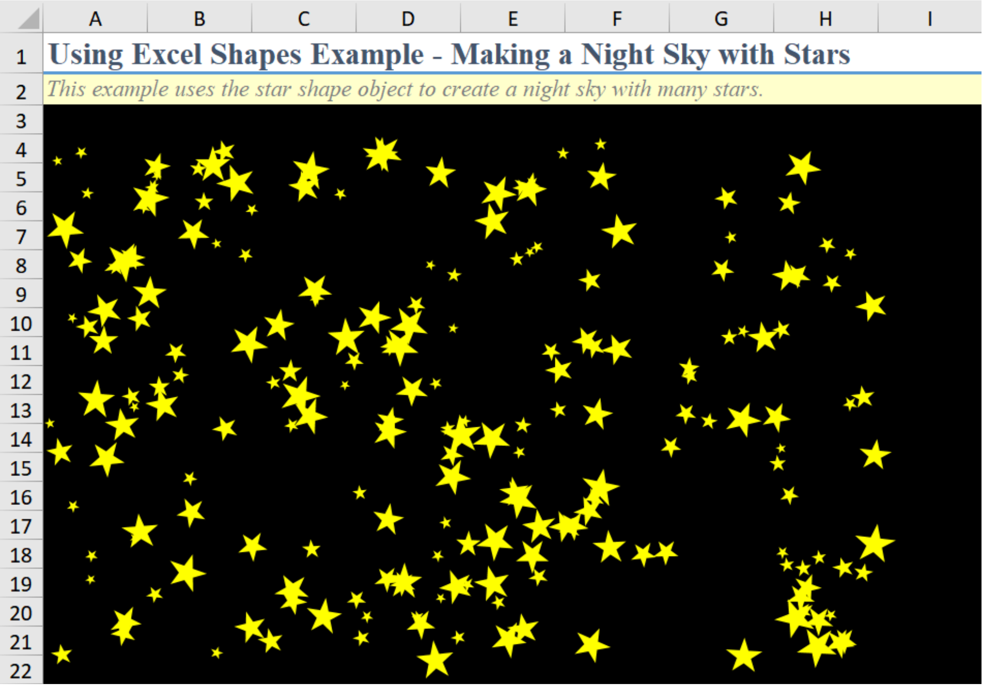
```
Face.Rotation = 60
```



Rotate
60 degrees

A Night Sky with 200 Stars

- In the next example, many stars with different size and orientation are created inside a worksheet
- Each star is created using several random numbers with different ranges
- An example of the worksheet is shown on the right:



Using Excel Shapes Example - Making a Night Sky with Stars

This example uses the star shape object to create a night sky with many stars.

Random Numbers for Each Star

- Each star uses these random numbers:
  - X position, an integer from 0 to 399 inclusive

    ```
    X = Int(Rnd() * 400)
    ```

  - Y position , an integer from 50 to 299 inclusive

    ```
    Y = Int(Rnd() * 250) + 50
    ```

  - Size, an integer from 5 to 19 inclusive

    ```
    Size = Int(Rnd() * 15) + 5
    ```

  - Rotation, a real number from 0 to 359.99

    ```
    Star.Rotation = Rnd() * 360
    ```

Generating The Stars

```
Dim X As Integer, _
    Y As Integer, Size As Integer
Dim Star As Shape

Randomize

Do While Worksheets("Night Sky").Shapes.Count < 200
  X = Int(Rnd() * 400)
  Y = Int(Rnd() * 250) + 50
  Size = Int(Rnd() * 15) + 5

  Set Star = ActiveSheet.Shapes.AddShape(
          msoShape5pointStar, X, Y, Size, Size)
  Star.Fill.ForeColor.RGB = vbYellow
  Star.Line.Visible = False

  Star.Rotation = Rnd() * 360
Loop
```

The loop runs 200 times

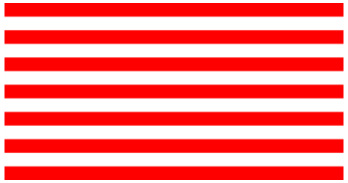
A yellow star without any border is put in the worksheet

The newly created star is randomly rotated

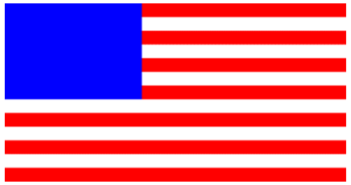
Drawing the American Flag 1/2

- The last example draws the American flag in five steps:

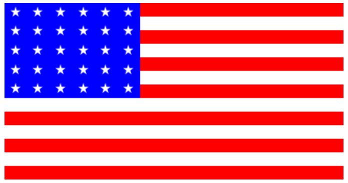
1. Use a loop to draw the red and white strips



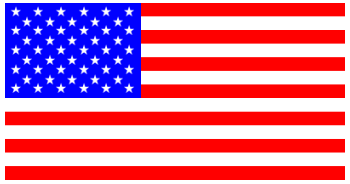
2. Draw the blue background



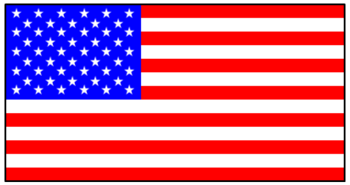
3. Use a nested loop to draw 5 rows of white stars with each row having 6 stars



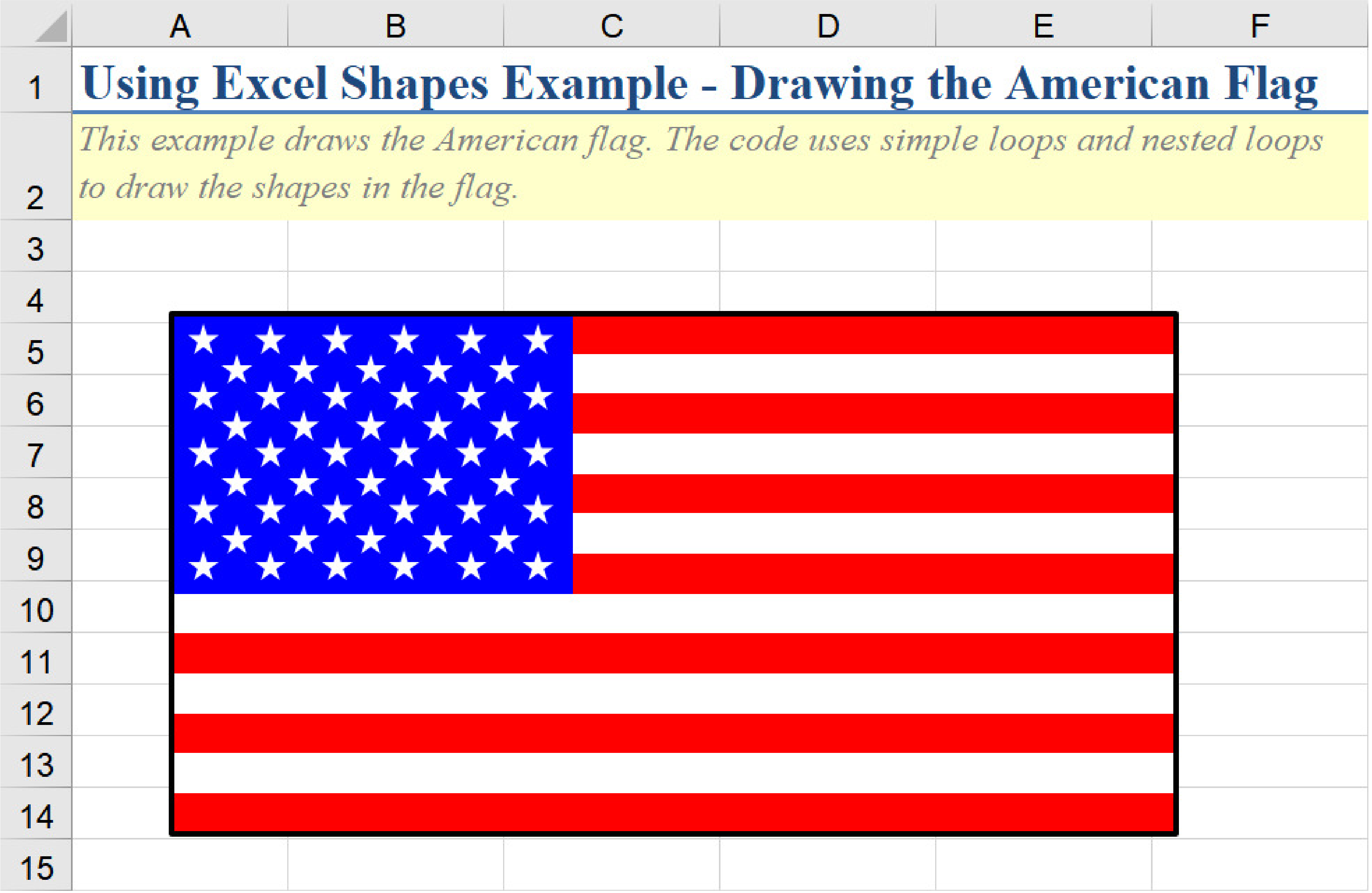
4. Use a nested loop to draw 4 rows of white stars with each row having 5 stars



5. Finally, draw a border around the flag



Drawing the American Flag 2/2



Using Excel Shapes Example - Drawing the American Flag

This example draws the American flag. The code uses simple loops and nested loops to draw the shapes in the flag.

Initialising the Variables 1/2

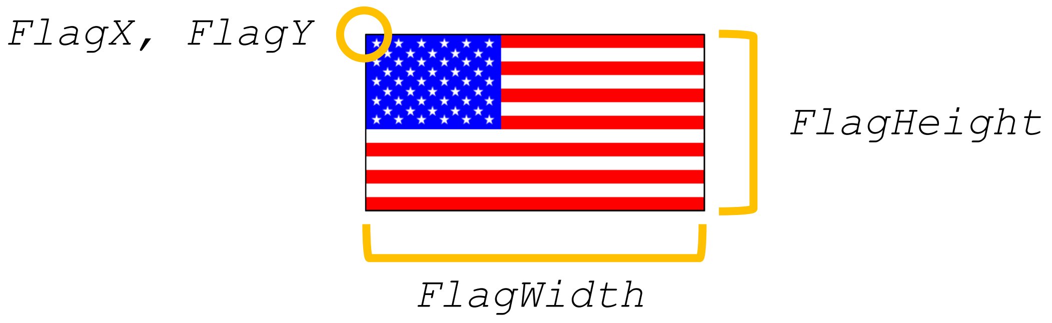
- Before drawing the flag, some variables are created at the start of the code

```
Dim FlagX As Integer, FlagY As Integer
Dim FlagWidth As Integer, FlagHeight As Integer
Dim UnionWidth As Integer, UnionHeight As Integer
Dim StarSize As Integer, StarSpacingX As Integer, _
    StarSpacingY As Integer
```

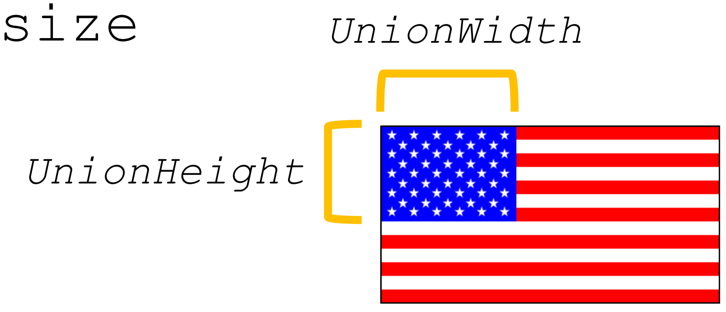
- These variables define the position and size of each component of the flag, as shown on the next slide

Initialising the Variables 1/2

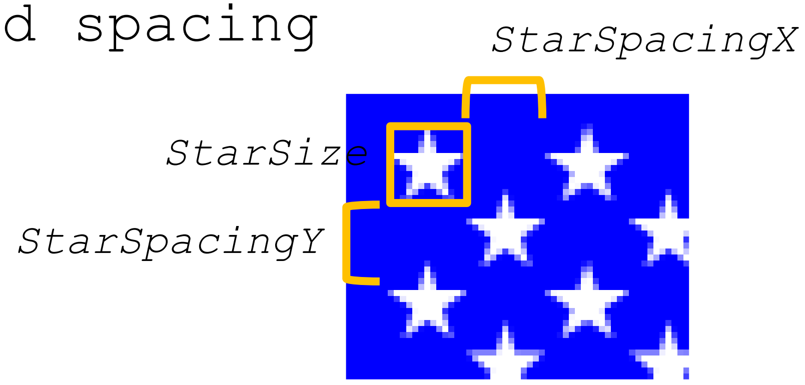
```
' Define the flag position and size
FlagX = 30
FlagY = 80
FlagWidth = 300
FlagHeight = 156
```



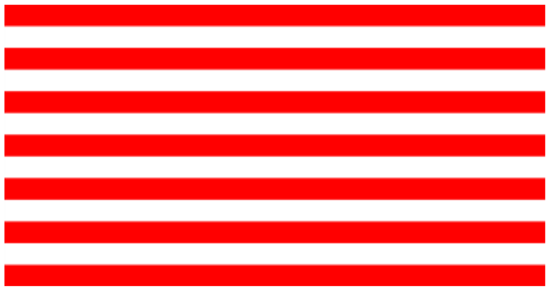
```
' Define the blue area size
UnionWidth = 120
UnionHeight = 84
```



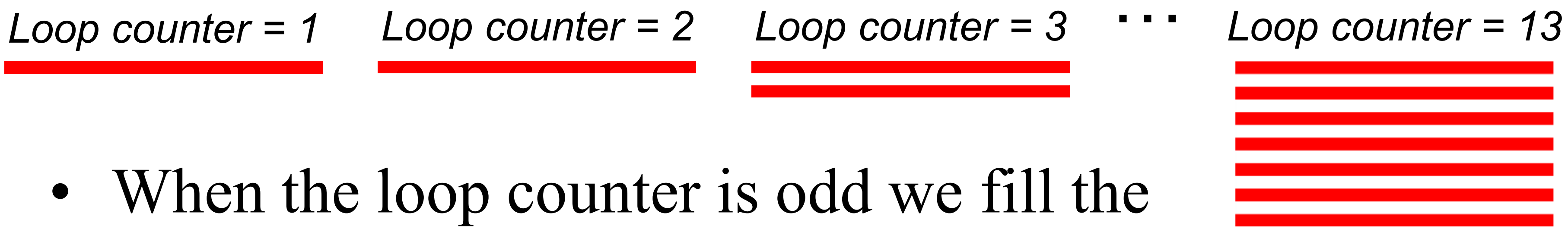
```
' Define the star size and spacing
StarSize = 9
StarSpacingX = 20
StarSpacingY = 17
```



Drawing the Strips 1/2



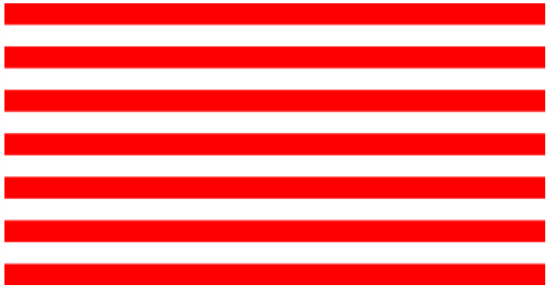
- Let’s draw the flag one step at a time
- First we will draw the stripes at the back
- There are a total of thirteen red and white strips
- We can use a for loop, which runs thirteen times, to draw thirteen strips, like this:



Loop counter = 1 *Loop counter = 2* *Loop counter = 3* . . . *Loop counter = 13*

- When the loop counter is odd we fill the stripe with white, otherwise, we fill the stripe with red

Drawing the Strips 2/2



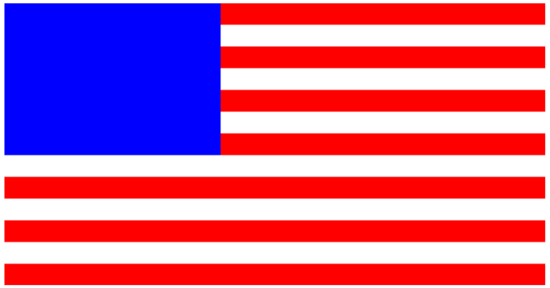
```
' Step 1 - draw the red and white strips
StripY = FlagY
For StripNo = 1 To 13
    ' Draw a rectangle for the strip
    Set Strip = ActiveSheet.Shapes.AddShape( _
        msoShapeRectangle, FlagX, StripY, _
        FlagWidth, FlagHeight / 13)

    ' Set the alternate colour of the strip
    Strip.Line.Visible = False
    If StripNo Mod 2 = 0 Then
        Strip.Fill.ForeColor.RGB = vbWhite
    Else
        Strip.Fill.ForeColor.RGB = vbRed
    End If
    StripY = StripY + FlagHeight / 13
Next StripNo
```

The loop runs thirteen times

Draw one strip

Set the fill colour of the shape based on the strip count

Drawing the Blue Background



- Drawing the blue area does not require the use of loops because it is a simple blue rectangle:

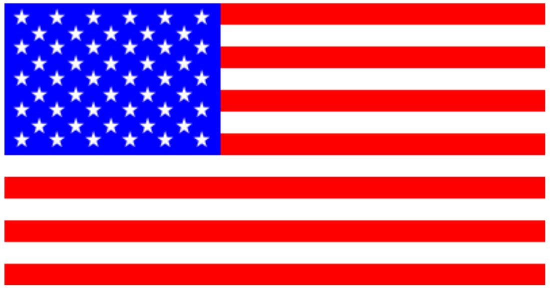
```
' Step 2 - draw the blue area

' Draw the blue rectangle
Set Union = ActiveSheet.Shapes.AddShape( _
    msoShapeRectangle, FlagX, FlagY, _
    UnionWidth, UnionHeight)
```

Draw a rectangle for the blue area

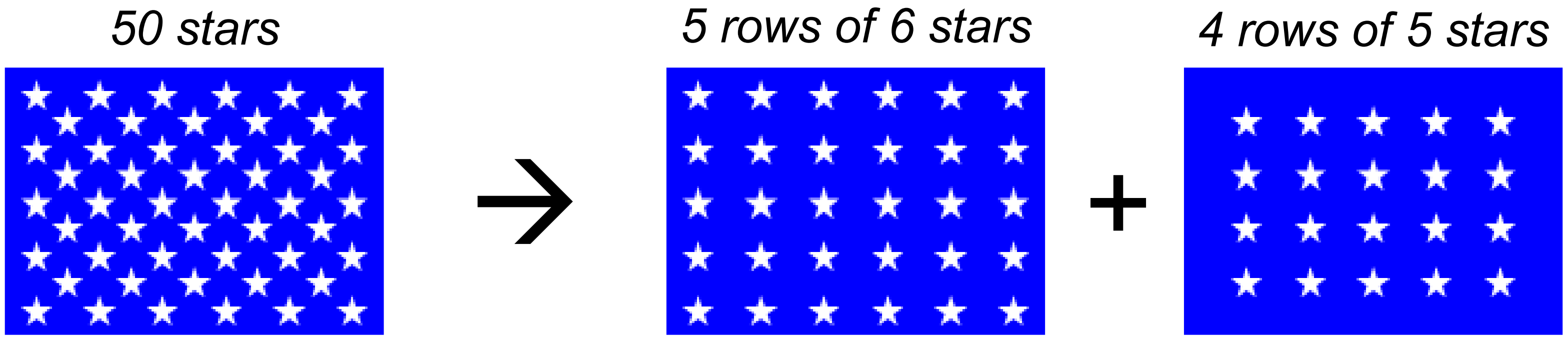
```
Union.Line.Visible = False
Union.Fill.ForeColor.RGB = vbBlue
```

The area is filled with blue without any border

Drawing the Stars 1/3

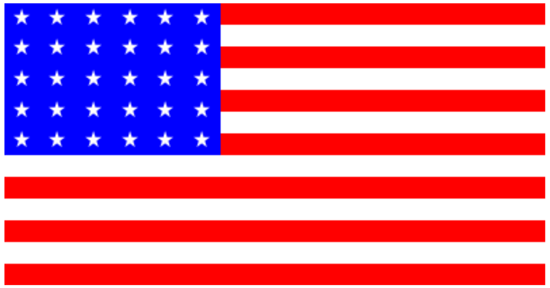


- Then we will draw 50 stars on top of the blue area
- Looking at the arrangement of the stars notice that they can be separated into two groups:



- Each of the groups can be drawn using a nested loop
- The outer loop handles the rows whereas the inner loop handles the columns

Drawing the Stars 2/3

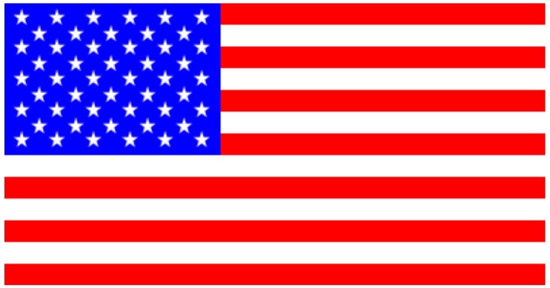


```
' Step 3 - draw the 5 rows of 6 stars
StarY = FlagY + 3
For Row = 1 To 5
    StarX = FlagX + 5
    For Col = 1 To 6
        ' Draw the white star
        Set Star = ActiveSheet.Shapes.AddShape( _
            msoShape5pointStar, StarX, StarY, _
            StarSize, StarSize)
        Star.Line.Visible = False
        Star.Fill.ForeColor.RGB = vbWhite

        StarX = StarX + StarSpacingX
    Next Col
    StarY = StarY + StarSpacingY
Next Row
```

Draw 5 rows of 6 stars

Draw a solid white star

Drawing the Stars 3/3



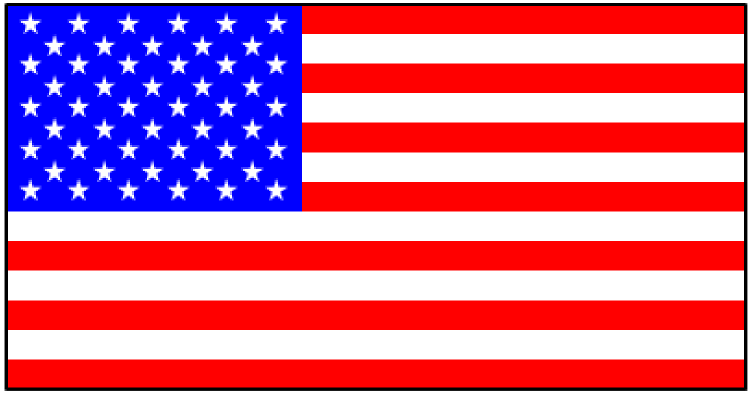
```
' Step 4 - draw the 4 rows of 5 stars
StarY = FlagY + 12
For Row = 1 To 4
    StarX = FlagX + 15
    For Col = 1 To 5
        ' Draw the white star
        Set Star = ActiveSheet.Shapes.AddShape( _
            msoShape5pointStar, StarX, StarY, _
            StarSize, StarSize)
        Star.Line.Visible = False
        Star.Fill.ForeColor.RGB = vbWhite

        StarX = StarX + StarSpacingX
    Next Col
    StarY = StarY + StarSpacingY
Next Row
```

Draw 4 rows
of 5 stars

Drawing the Border

- Finally, we draw the flag border using an unfilled rectangle:

```
' Step 5 - draw the flag border

' Draw the border of black colour
Set Flag = ActiveSheet.Shapes.AddShape( _
    msoShapeRectangle, FlagX, FlagY, _
    FlagWidth, FlagHeight)
Flag.Fill.Visible = False
Flag.Line.ForeColor.RGB = _
    vbBlack
Flag.Line.Weight = 1.5
```



Specify a black border with a thickness of 1.5 for the shape