

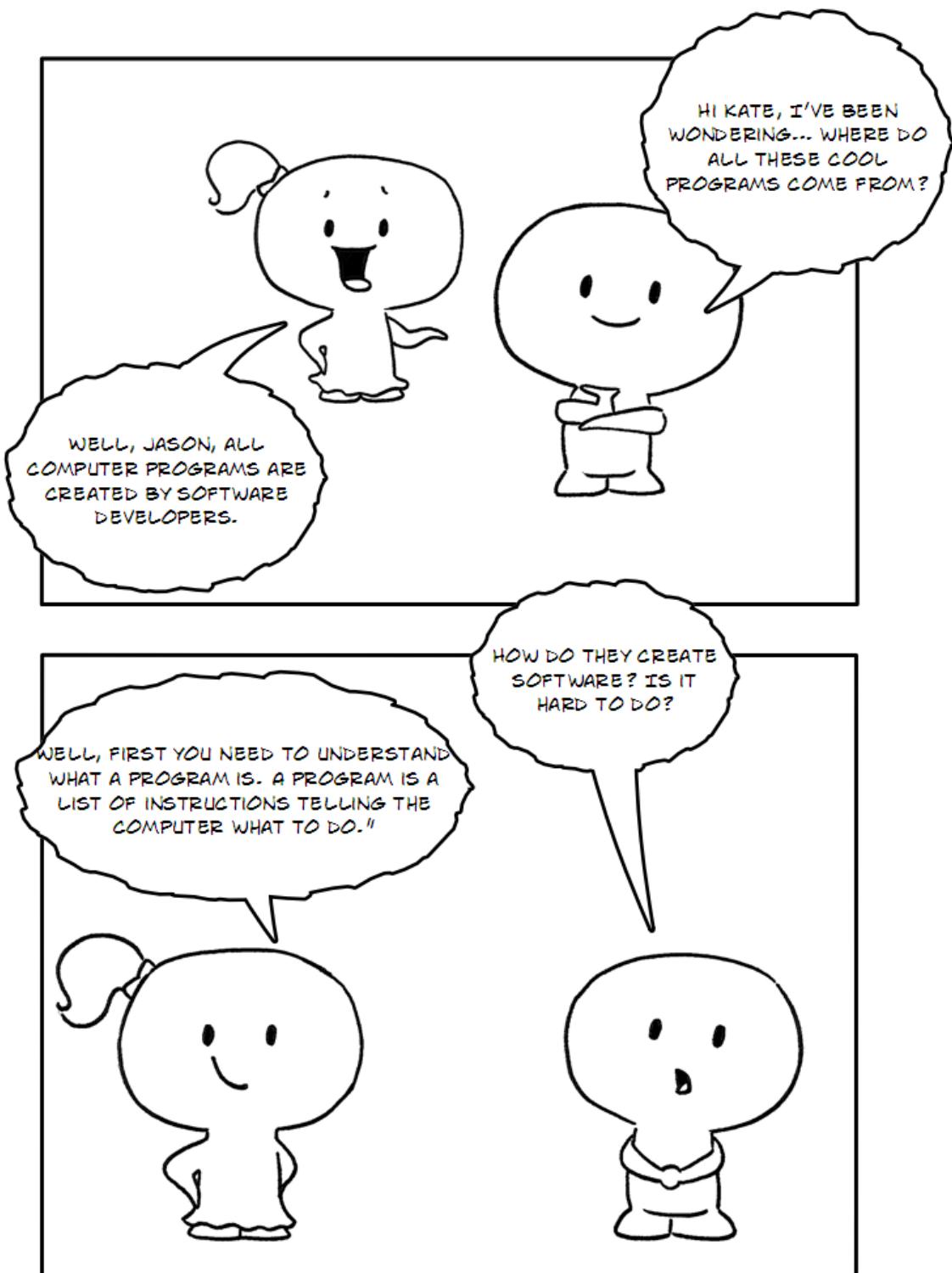
# *Chapter 1*

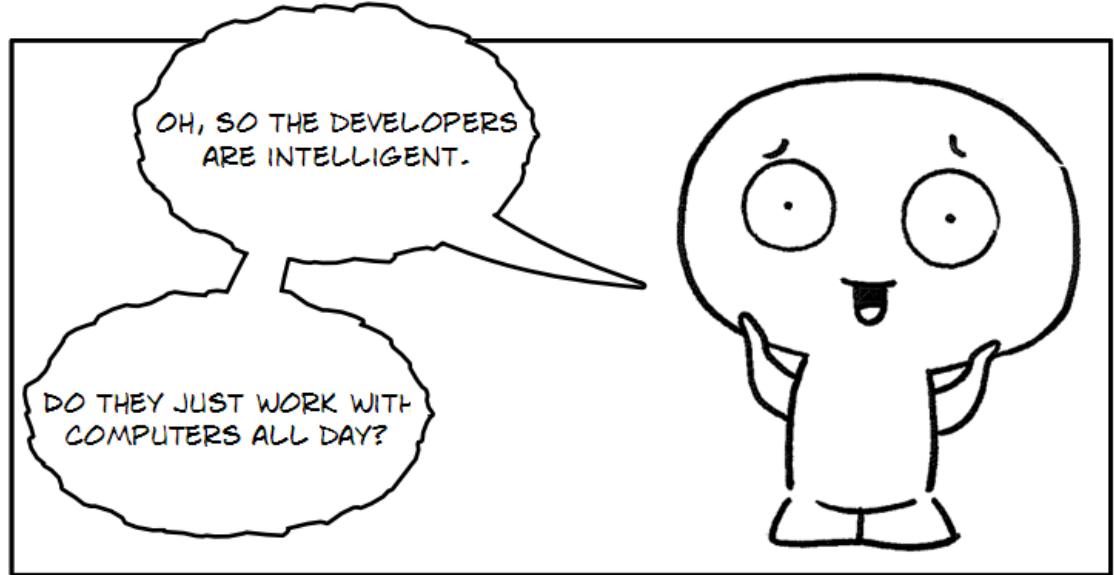
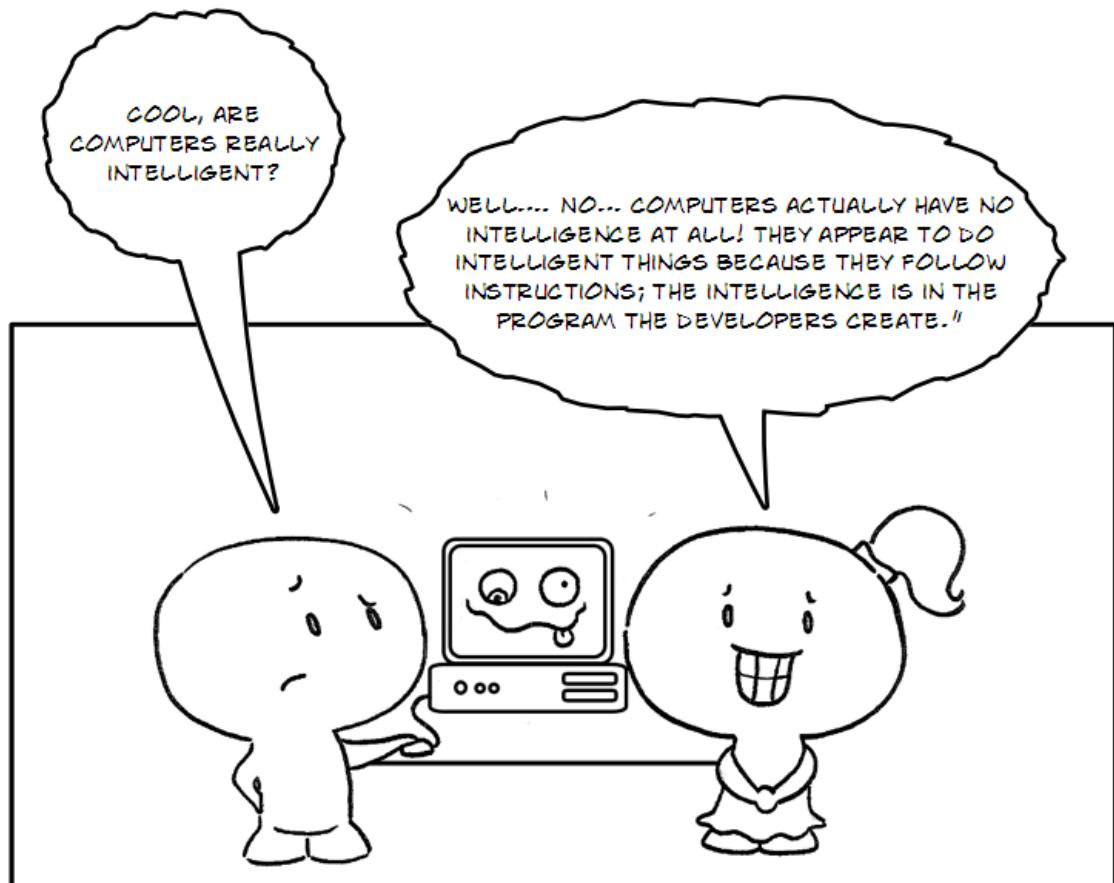
*"Hello World"*

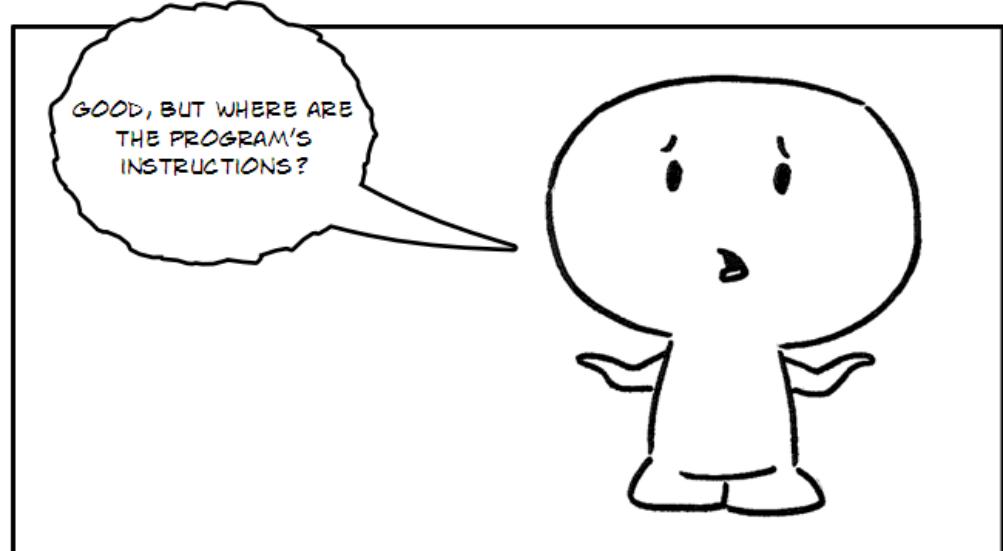
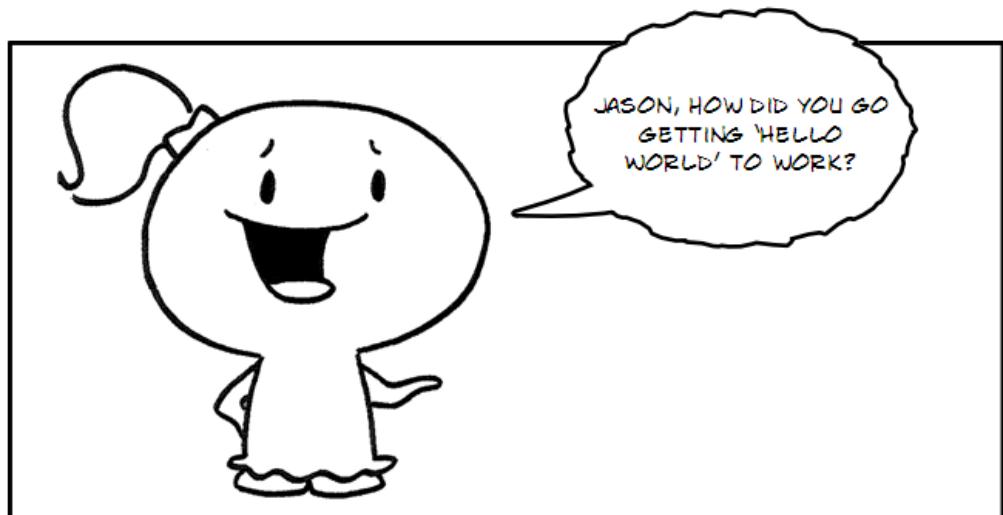
## Summary:

In this chapter you will perform basic actions creating a “Hello World” program. You will be changing the color and position of elements on the screen, and drawing a smiley face.

To start you need to have Visual Studio, which has to be set up for Visual Basic .NET, a template that has been provided for this project and it can also be downloaded from [www.swingame.com](http://www.swingame.com) (Download – Swingame SDK – Visual Studio 2008). The template should be copied into Documents -> Visual Studio 2005/2008 -> Templates -> Project Templates.







**Part 1**

"Hello World" program is the program you create when you start learning any programming language. It is a simple program which prints the "Hello World!" message on the screen. It ensures everything is functioning correctly with your system before you begin anything complex. It also tells you whether the SwinGame project set up correctly. Follow these steps to get the "Hello World" program working with SwinGame.

1. Open Visual Studio, and click File > New > Project. As shown in Figure 1.

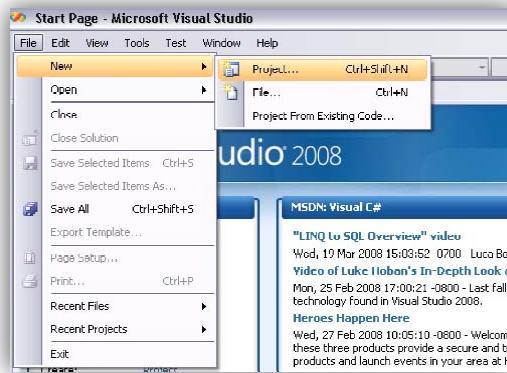


Figure 1

2. Select the Visual Basic/Basic Language and click the SwinGame VB.NET Project Template. Name new project as "HelloWorld". Click OK. The process is shown in Figure 2.

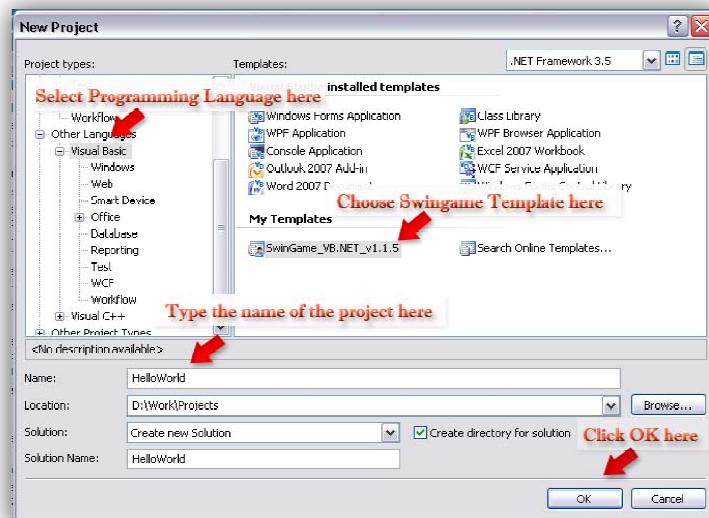
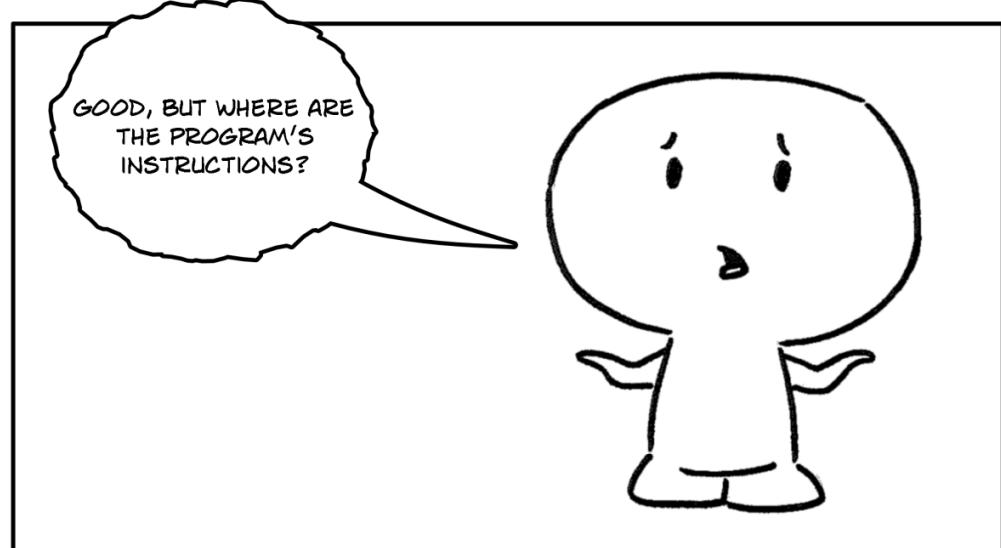
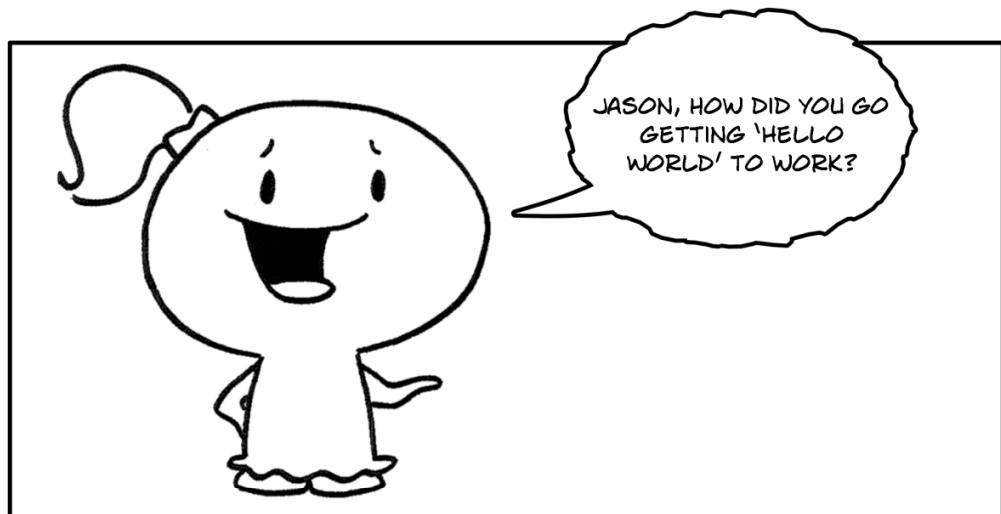
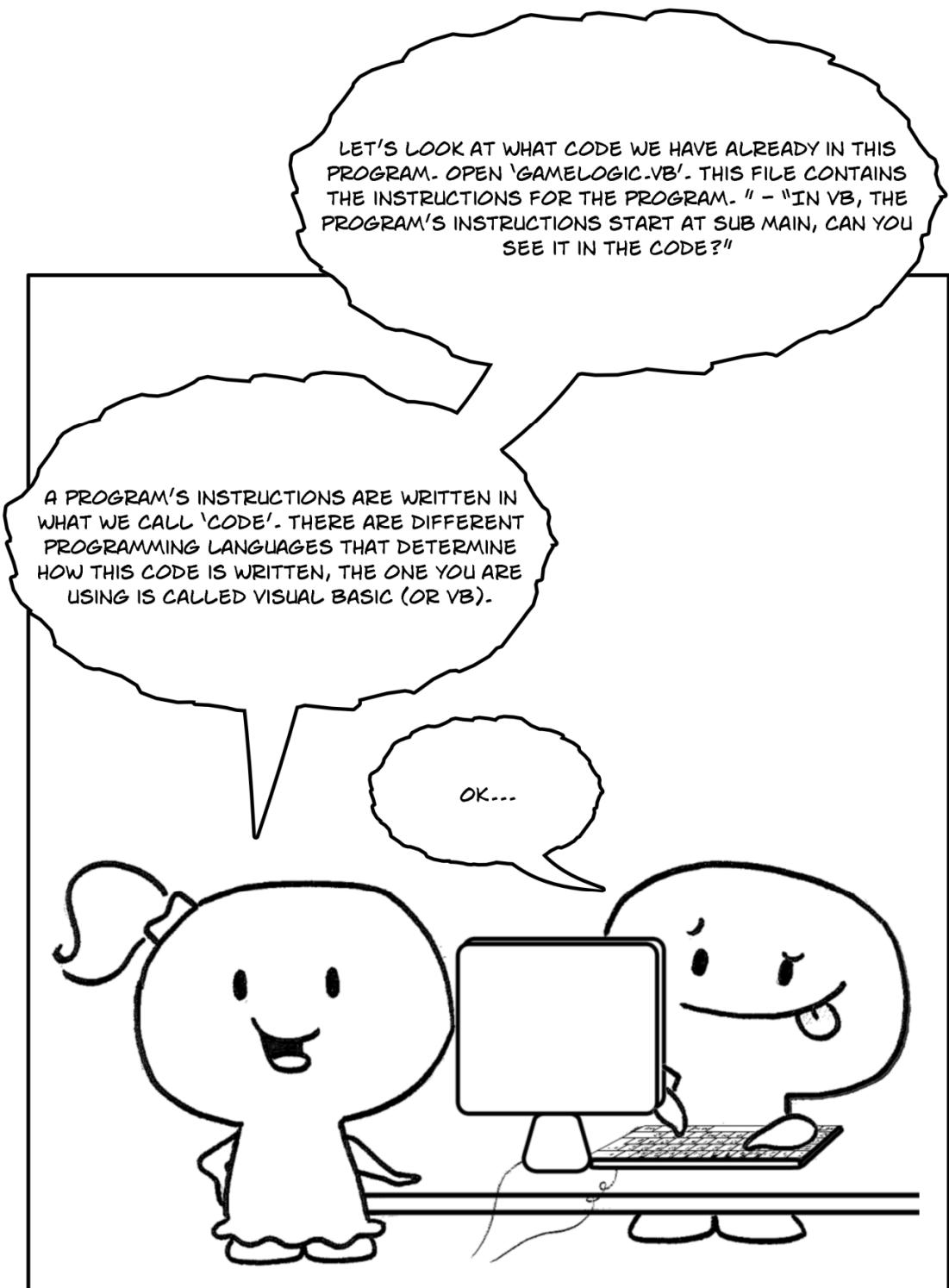


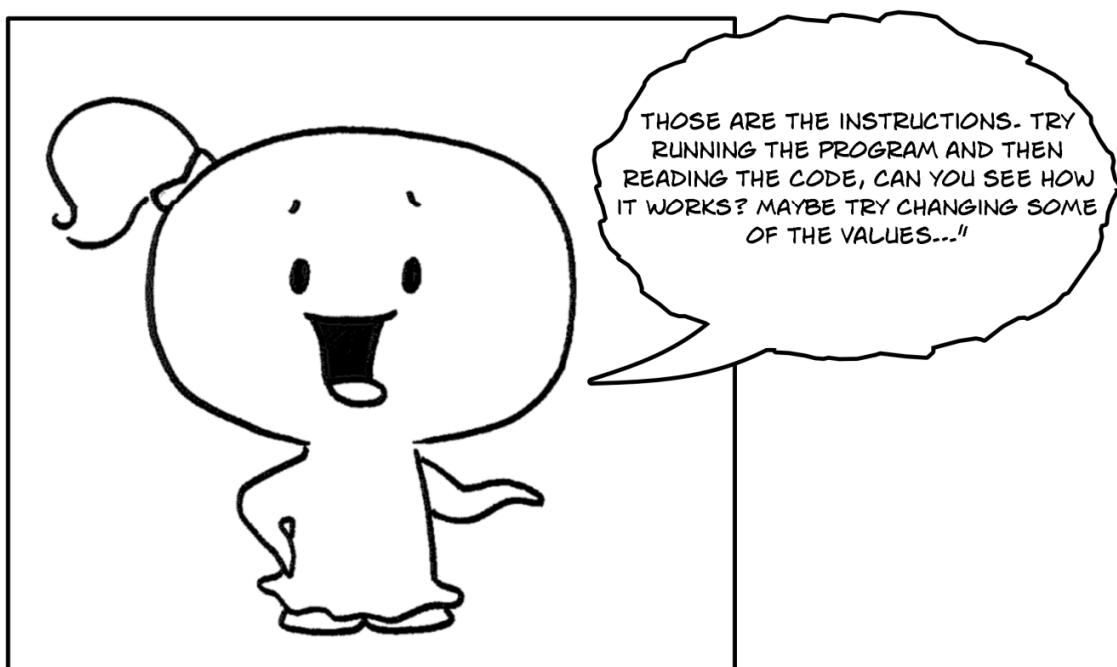
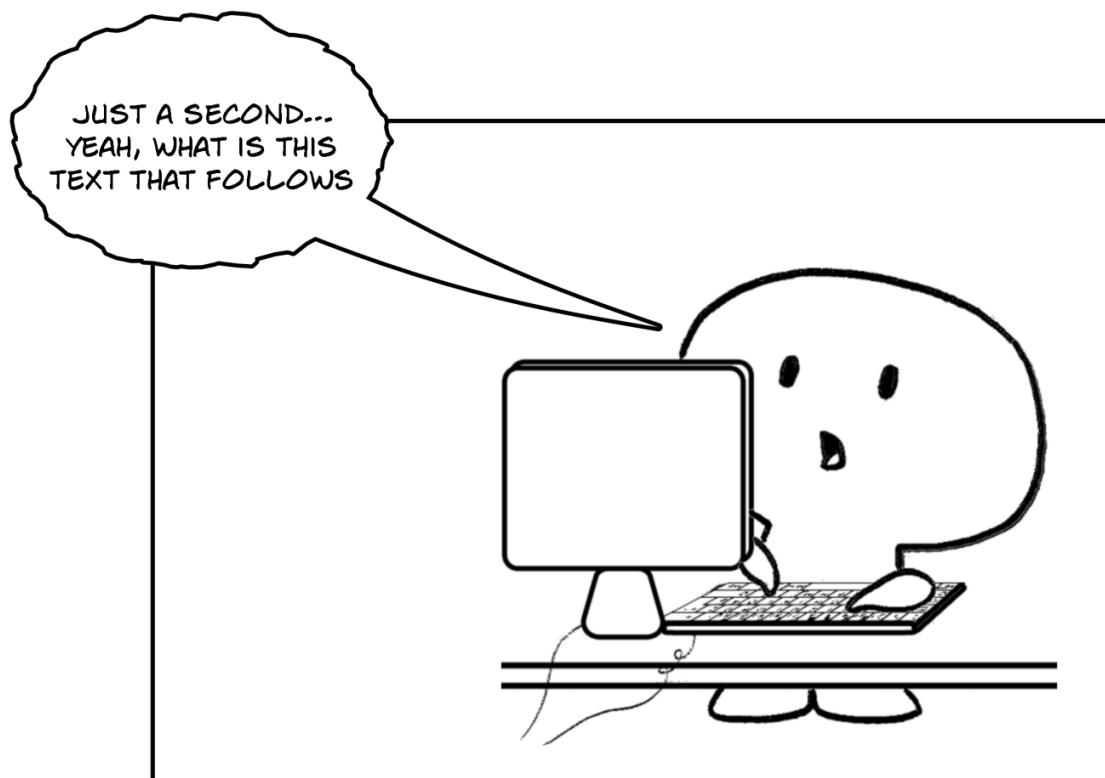
Figure 2

3. Press the "StartDebugging" button at the top of the screen (it looks like a green arrow , F5 works too) to see what it does then close the window.

 *Question 1. What is happening on your SwinGame screen? Answer on the worksheet.*







**Part 2**

In the Solution Explorer on the right hand side of the screen double click on "GameLogic", as shown in Figure 3. You should be presented with the following code:

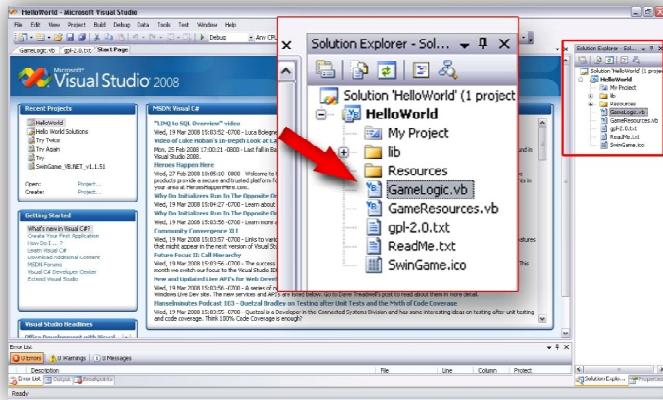


Figure 3

Code shown:

```

Module GameLogic
    Public Sub Main()
        'Opens a new Graphics Window
        Core.OpenGraphicsWindow("Game", 800, 600)

        'Open Audio Device
        Audio.OpenAudio()

        'Load Resources
        LoadResources()

        'Game Loop
        Do
            'Clears the Screen to Black
            SwinGame.Graphics.ClearScreen()

            'Draws rectangle on the screen
            Graphics.FillRectangle(Color.Red, 20, 150, 500, 50)

            'Draws text on the screen
            Text.DrawText("Hello World!", Color.Aqua, GameFont("ArialLarge"), 50, 50)

            'Refreshes the Screen and Processes Input Events
            Core.RefreshScreen()
            Core.ProcessEvents()

        Loop Until SwinGame.Core.WindowCloseRequested() = True

        'Free Resources and Close Audio, to end the program.
        FreeResources()
        Audio.CloseAudio()

    End Sub

End Module
  
```

The purpose of this code is to create a starting point for your project. If you press the "StartDebugging" button again at the top of the screen (looks like a green arrow , F5 works too) you can hopefully see better how the code works. Have another look, and then close the window. You can basically read what it is doing:

*'Comments are in green with a single quotation at the start like this sentence.*  
 Comments are just notes made by the programmer to inform anyone looking at the code what it does; they don't affect the code at all. You can also use them by putting them in front of a line of code that is not working probably so that it does not execute.

Every line in the program has a purpose, but the most important parts are subs (i.e. `Public Sub Main()... End Sub`) and sub calls (i.e. `Graphics.FillRectangle(rectangle color, Xpos, Ypos, Width, Height)` or `Text.DrawText("Text To Draw", Color, GameFont("FontName"), Xpos, Ypos)`). We will learn more about them later.

The Game Loop is a piece of code that keeps repeating its execution over and over again until the game window is closed. It is the actual "working" part of your game - it draws elements on the screen and provides the main functionality of the game. In our case, the game loop starts from "`Do`" keyword followed by some code to execute continuously and ends "`Loop Until SwinGame.Core.WindowCloseRequested() = True`", this tells the program to stop executing the code when the window is closed.

The "Do loop" in VB.NET keeps executing statements while or until a condition is true (in our loop the condition is when the user closes the window). The loop we are using has to be executed at least once because the test (Do Until) occurs at the end of the loop. Here is Nassi-Shneiderman (NS) Diagram for the game loop, you can see what is executed before the "Do Until" test at the end; see if you can understand what is going on:

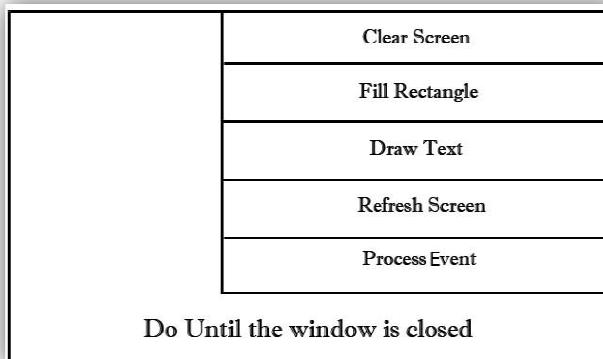


Figure 4. Game Loop

### Did you know:

A Nassi-Shneiderman diagram (or NSD) is a graphical design representation for structured programming. Developed in 1972 by Isaac Nassi and Ben Shneiderman, these diagrams are also called structograms, as they show a program's structures.



### Question 1:

- What purpose do comments serve in code?
- What symbol do you use to make something a comment?
- Why do you think we might use NS diagrams to represent code?
- What would happen to the loop the "Do Until" condition was never met?

**Exercise1: Reading the code**

*Answer the following questions on the worksheet:*

- Which line number contains the code that tells the computer to draw text on the screen?
- Which line number contains the code that tells the computer to draw a rectangle on the screen?
- Where (line numbers) does the game loop start and end?

**NOTE:** If there are no line numbers on the left side of the window, add them manually:

- (Visual Studio 2008) Go to tools -> Options (Figure 5)

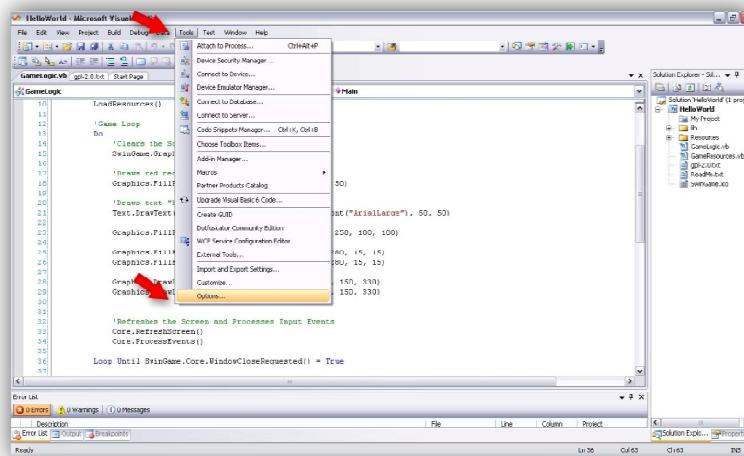


Figure 5

- Choose Text Editor -> All Languages -> General
- Check Display Section -> Line Numbers (Figure 6)

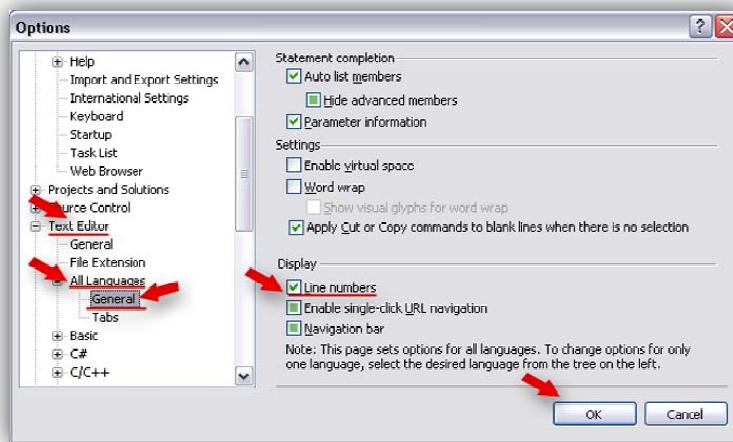


Figure 6

**Exercise 2: Changing the text**

Make the following changes in your *Hello World* program and write your solutions onto the worksheet:

- Change the text "Hello World!" to "Hello Your Name!"

*Hints:* When the mouse is over the sub call (i.e. LoadResources()), a pop up window will appear which contains the list of parameters that this function takes and their (Figure 7) this can help you later on when you are adding your own sub calls.

```
' Load Resources
LoadResources()
Public Sub LoadResources()
The Resources Class stores all of the Games Media Resources, such as Images, Fonts Sounds, Music, and Maps.
Do
    'Clears the Screen to Black
    SwingGame.Graphics.ClearScreen()

    'Draws red rectangle
```

Figure 7

**Exercise 3: Changing the color**

Make the following changes in your *Hello World* program and write your solutions onto the worksheet:

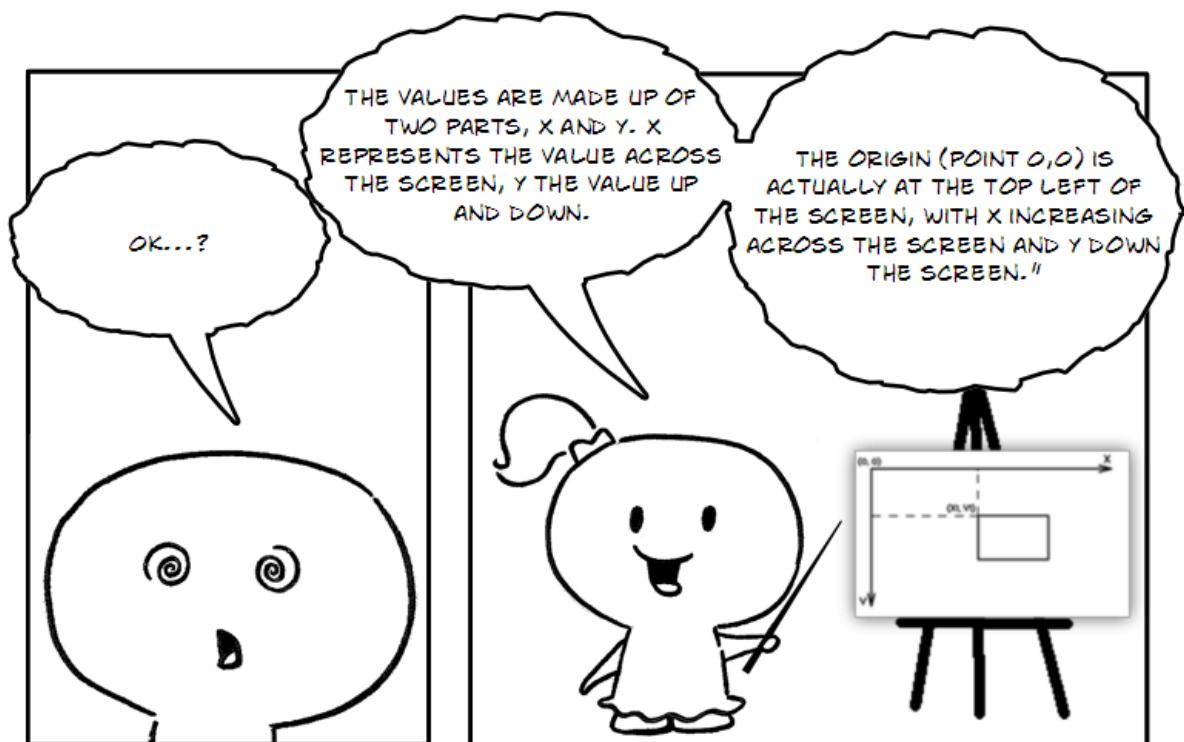
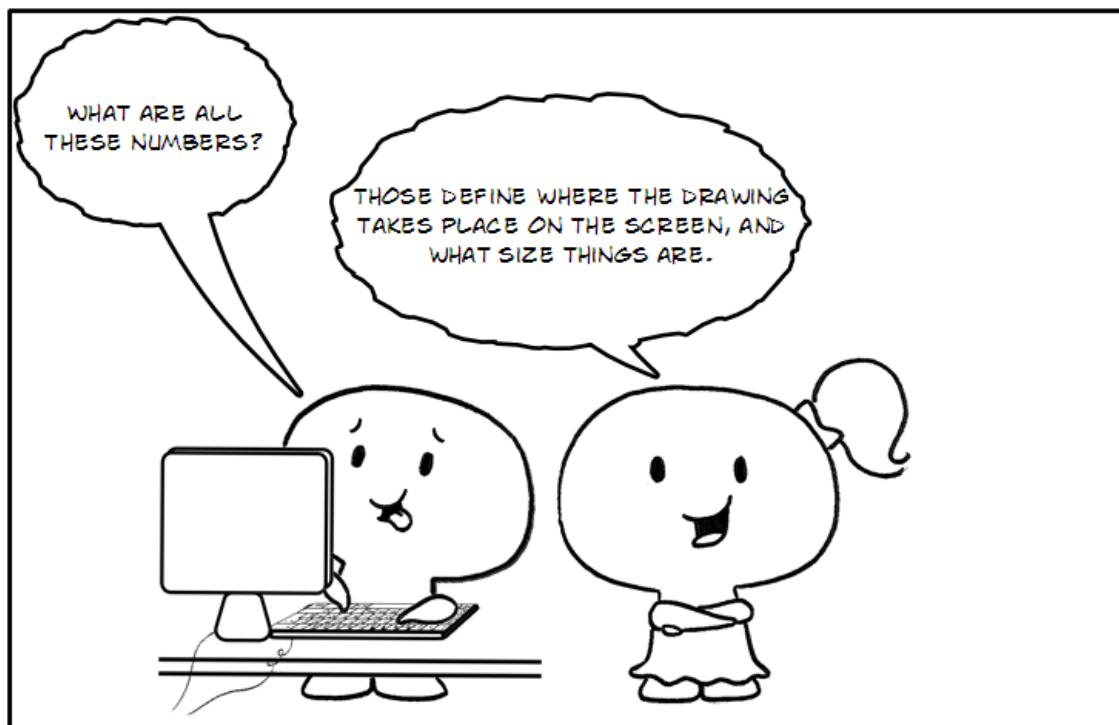
**NOTE:** Spelling is important. In this case, you MUST use American spelling - COLOR, not Colour. Otherwise, it will produce an error message and prevent your program from executing.

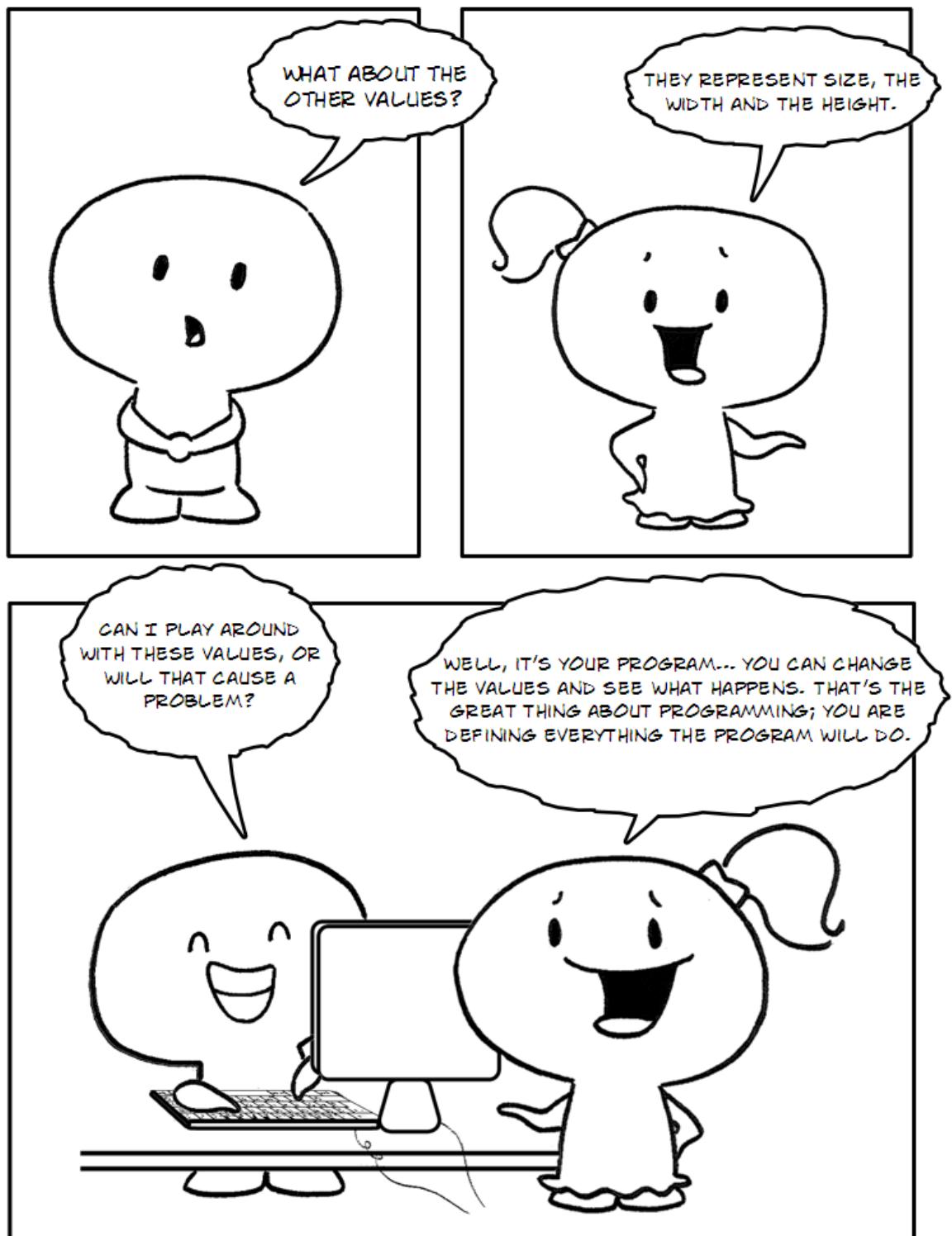
- Change the color of the text to Color.GreenYellow.

*Hints:* When you type Color. a list of all possible colors will appear.

- Change the color of the rectangle to Color.Blue.
- Change the background color of the screen to Color.White.

**NOTE:** The ClearScreen() sub clears the screen to black color by default. To customize the background color, simply type "Color." followed by color name SwingGame.Graphics.ClearScreen(Color.White).





**Part 3****Exercise 1: Locations on the screen**

Draw the following shapes and text onto your worksheet.

- On your worksheet, draw a small rectangle with coordinates  $X = 15$  and  $Y = 5$  by hand.
- Draw the text "Hello Your Name" at  $X = 5$ ,  $Y = 20$ .

*Hints:* The location of each element is determined by its X and Y coordinates. X and Y coordinates are taken from the top left corner of each element as shown in Figure 8.

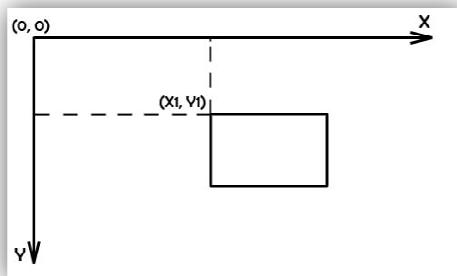


Figure 8. Coordinates

**Exercise 2: Changing drawing locations**

Make the following changes to your *Hello World* program and write your solutions onto the worksheet:

- Change the location of the text; put it in the middle of the screen.
- Change the location of the rectangle; put it under the text.

**NOTE:** The size of SwinGame screen is 800 pixels wide (X Axis) and 600 pixels tall (Y Axis) (Figure 9) A pixel is basically a dot, your screen is made up of thousands of these dots so the SwinGame screen is 800 dots wide by 600 dots tall. These dots can be all different colors.

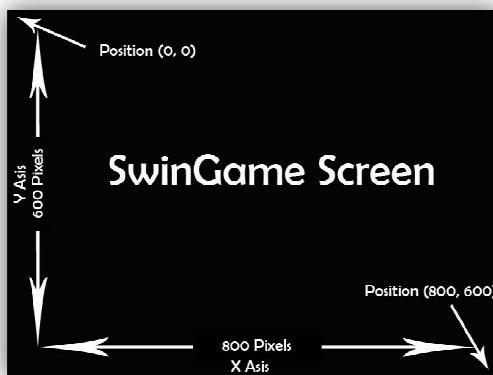


Figure 9

**Exercise 3: Size of an element**

Draw the following shapes and text onto your worksheet.

- a) Draw a rectangle with width = 5, height = 10 at the position X = 0, Y = 5 by hand.
- b) Draw a rectangle with width = 10, height = 5 at the position X = 5, Y = 10 by hand.

**Exercise 4: Draw an element with the new size.**

Make the following changes to your *Hello World* program and write your solutions onto the worksheet:

- a) Change the size of the rectangle to width = 630 and height = 20.

*Hints:* The size of any element, such as a rectangle or a circle, is declared by using width and height parameters (Figure 10). Remember objects are drawn starting from the top-left corner.

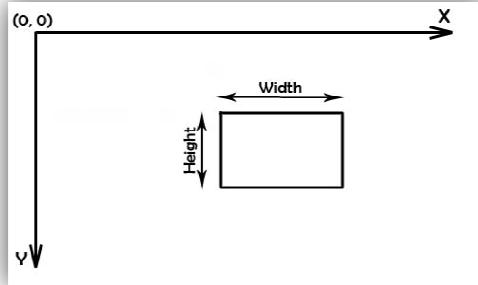
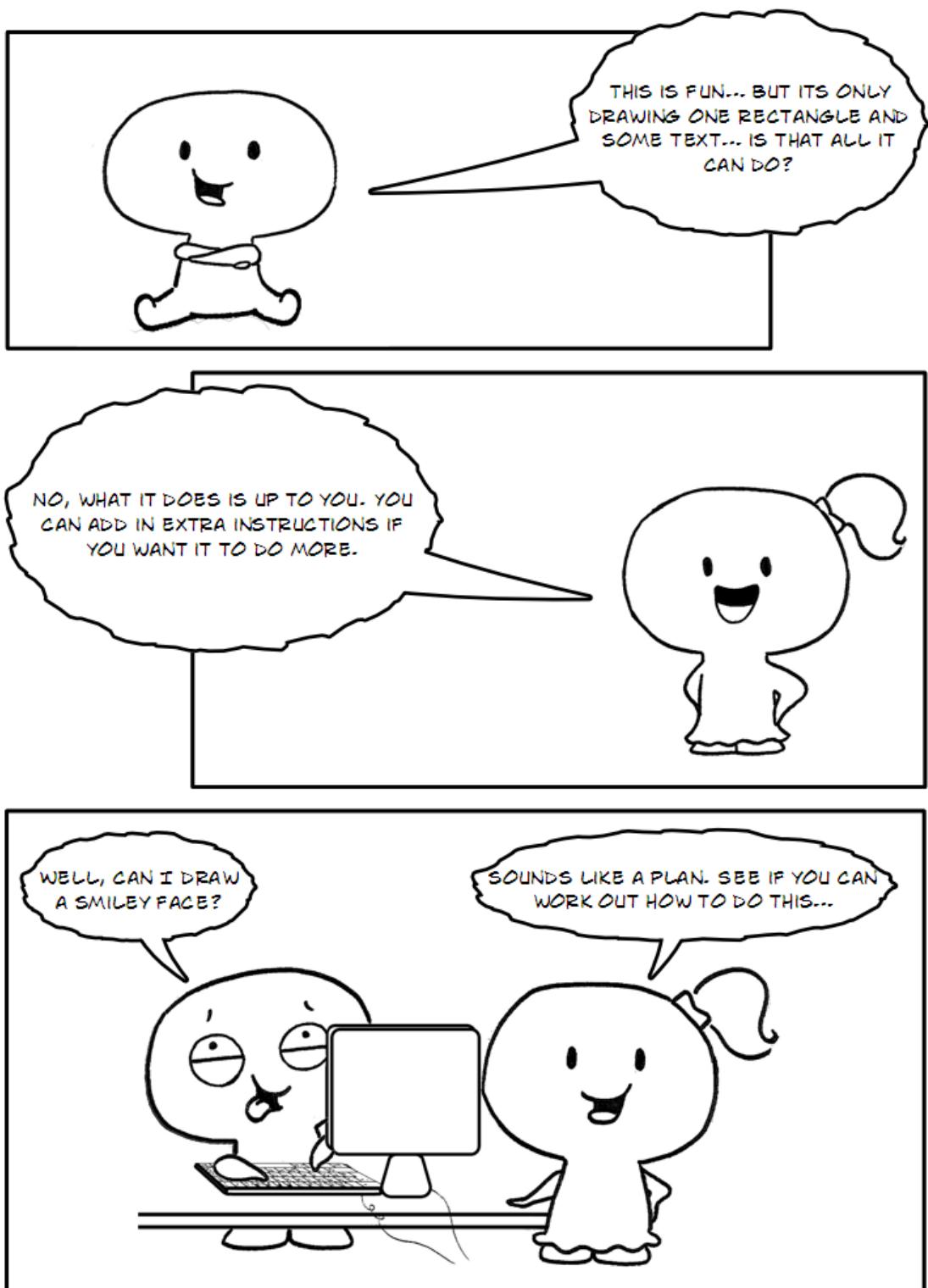


Figure 10. Size



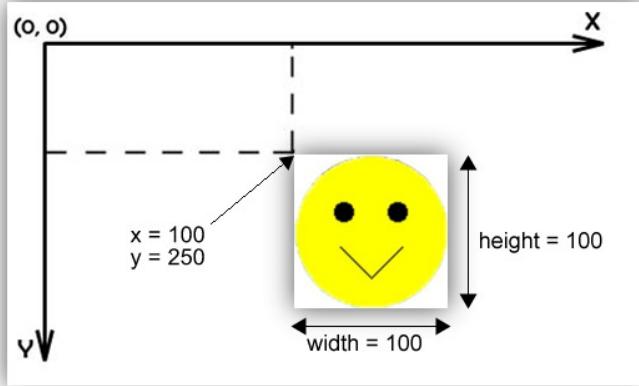
## Exercise 5



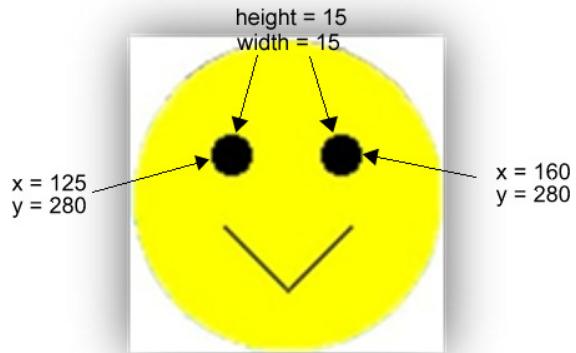
Draw "Smiley" face on the screen then put the code you entered to the worksheet:

- 1) Draw a yellow circle on the screen – "face". Use:

Graphics.FillEllipseOnScreen(Color, Xpos, Ypos, Width, Height) and the following dimensions:



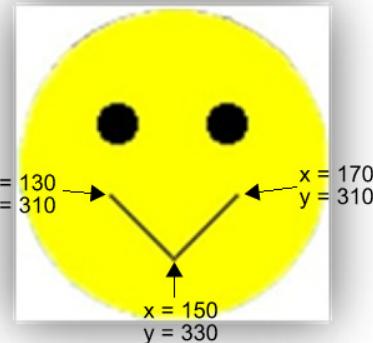
- 2) Draw the eyes – two black circles inside the yellow circle. Use the same sub call to draw, i.e.:



- 3) Draw the "smile" on the screen.

Tell the computer to draw two lines which are connected at the bottom, inside the yellow circle. To do so, use:

```
Graphics.DrawLineOnScreen(Color, XPosStart, YPosStart, XPosEnd, YPosEnd)  
i.e.:
```



**NOTE:** There is no arc line in SwinGame so the "smile" is made by two lines.

### IMPORTANT!!!

 Remember to save your exercise where you can find it again, often in programming you will want to use some bit of code you made before and it is frustrating if you can't find it. This file you should have named "Hello World" and it will be saved by default in "My Documents/Visual Studio 2008/Projects/HelloWorld". If you want to open this file again open the "HelloWorld.sln" file as this is the "solution" file and will open the whole project (not just parts of it). Once you have finished then you can close Visual Studio or move on to the next chapter.