# Task 3 - Extending the ShapeParser

Prepared by William Reid
September 18, 2014

# 1 Modifying Class Functions

## 1.1 Shape Class - (4 marks)

Create a function called *print()* that returns a string containing the information of the shape. It should be in the same format as it was read in:

*type: (X1, Y1), (X2, Y2), ... (Xn, Yn), colour*

## 1.2 DrawPanel Class - (2 marks)

Create a function (name of your choosing) that returns the list of shapes that the DrawPanel class has stored locally.

## 1.3 DrawFrame Class - (4 marks)

Modify the constructor to get all the shapes stored in the DrawPanel and send them to the ShapeParser's *writeFile()* method **(2 marks)**. Be careful where you place this code as incorrectly placing the function calls will cause the program to fail.

Verify this works by adding code to the *writeFile()* method that calls (and prints out the returned string) the *print()* method we made earlier for each shape **(2 marks)**.

# 2 Writing to File - (7 marks)

Now that we can get the information of each shape, we want to write it out to a file. This can be done using Java's **BufferedWriter** class. The following resource provides a good example of how to implement the required features:

http://www.mkyong.com/java/how-to-write-to-file-in-java-bufferedwriter-example/

## 2.1 Checking the Output - (2 marks)

If you have correctly implemented the write methods, when you run the code you should obtain a new text file (that you may name whatever you wish) that looks identical to the sample file provided to you.

## 3 Choosing the File to Open/Save - (8 marks)

Modify the code in the DrawFrame constructor that creates the ShapeParser object to assign it to a local variable, such that it can be accessed anywhere in the class. Do the same for the creation of the DrawPanel object and also remove any other code that calls the read/write functions of the ShapeParser **(2 marks)**.

Modify the private listener classes that *Open* and *Save* files such that when these operations occur they:

- Open - pass the selected file to the *readFile()* method of the ShapeParser object, then pass the returned list of shapes to the DrawPanel object **(3 marks)**.
- Save - retrieves the shapes from the DrawPanel object and then passes them to the *writeFile()* method of the ShapeParser object. The output file must be named "shapes_output.txt" **(3 marks)**.

## 4 Testing - (10 marks)

Thoroughly test the new additions you have made. You may need to add additional functions and/or logic to ensure the program always behaves correctly. I will extensively test your program and attempt to break it, so ensure it does not crash when asked to do silly things (like write out the shapes contained in DrawPanel, but no shapes have been added).

## 5 Checking your Code (8 marks)

**Note: For each file - 1 mark each goes to comments/layout and conforming to the function/variable naming standards. This does not include Draw.java as this has not been changed.*

Review your code for layout, structure, comments and how it measures against the supplied standards. When you have completed this check, ensure it is uploaded to the Git repository (you should be committing your work once you have completed each class).

Notify me that you have finished and I will check your work, giving feedback on how things can be improved, including your total mark out of **45**. After I have given feedback you may be asked to modify the code (based on my comments). Once you have completed this and notified me, will move on to the next task. Good luck!