CC3 – Object Oriented Programming

Laboratory Exercise #8
Graphics Class

Name: _____ Date: _____

Code/Schedule: _____ Terminal #: _____

Topic(s) Covered: AWT package, Graphics class, paint() method

Estimated Completion Time: 2 meetings

Objectives:

1. To utilize the predefined methods found in the Graphics class under the AWT package.
2. To demonstrate creativity in designing graphics for a certain class.

Activity:

After practising with the HappyFace demo activity, write a program that shows your own RPG character (e.g., Wizard) using the different shapes found in the Graphics class. The program should ask for a name and asks the user to choose from a list of colors using JOptionPane. The name will appear in the Graphics canvas on top of the RPG character, while the color will be applied to the character's main parts (other parts may have colors other than the inputted color).

Required: There exists at least one or more of each of the following methods:
- drawOval()/fillOval()
- drawRect()/fillRect()
- drawPolygon()/fillPolygon()
- drawLine()
- drawArc()/fillArc()
- setColor(new Color(R,G,B))
- setFont(new Font(type, style, size))
- drawstring()

Laboratory Exercise Score Sheet

Trait	Exceptional (4)	Acceptable (3)	Amateur (2)	Unsatisfactory (1)
Specifications	The program works and meets all of the specifications.	The program works and produces the correct results and displays them correctly. It also meets most of the other specifications.	The program produces correct results but does not display them correctly.	The program is producing incorrect results.
Readability	The code is exceptionally well organized and very easy to follow.	The code is fairly easy to read.	The code is readable only by someone who knows what it is supposed to be doing.	The code is poorly organized and very difficult to read.
Reusability	The code could be reused as a whole or each routine could be reused.	Most of the code could be reused in other programs.	Some parts of the code could be reused in other programs.	The code is not organized for reusability.
Documentation	The documentation is well written and clearly explains what the code is accomplishing and how.	The documentation consists of embedded comment and some simple header documentation that is somewhat useful in understanding the code.	The documentation is simply comments embedded in the code with some simple header comments separating routines.	The documentation is simply comments embedded in the code and does not help the reader understand the code.
Delivery	The program was delivered on time.	The program was delivered within a week of the due date.	The code was within 2 weeks of the due date.	The code was more than 2 weeks overdue.
Efficiency	The code is extremely efficient without sacrificing readability and understanding.	The code is fairly efficient without sacrificing readability and understanding.	The code is brute force and unnecessarily long.	The code is huge and appears to be patched together.