**Lab: Lab 1**

**Registration number: 1703055**

|  |
| --- |
| **Problem statement(s)**  The aim of this lab was to refresh my memory of the InteliJ IDE in the first exercise and to recap object orientated programming in the second & third task. In these two tasks we made use of dynamic binding with shapes and expansion of the classes to construct the particular shape objects with their relevant attributes.  Overall this lab was to introduce the structure of the labs, making use of GitHub and introducing the lab reports. |

|  |
| --- |
| **Program description(s)**  For the first exercise this was a very simple hello world program created within InteliJ IDE with a single class named ‘HelloWorld’ which has a main function within that outputs “Hello World”.  For the second exercise, the class must have for shape which is extended on by rectangle and Circle in order to be able to create objects of those types.  My solution has an abstract class of Shape where the main attributes are defined for this object as the class variables covering name and colour. This class is then extended in Circle, Rectangle and RedRectangle classes. In each of these classes further attributes are defined for both these particular shapes within the constructor; attributes include the centre values and the length/width for rectangles or circumference for circles.  Then there is a *draw*() method within each class to print a description about the particular shape object. Here all the attributes are printed out to the user about the shape  Finally in the *main*() class a new shape instance [object] is created with new for each shape created in the array. Where the correct information if outputted to the user as the *printName*() and *draw*() function is called for each object.  For the final exercise I completed the employee task an abstract class of employee where the main attributes are defined for the such as name, salary, dofB. This class was then extended in development, managers and sales classes. In each of these classes further attributes are defined for both these roles by using constructors; attributes include number of sales for employees and more.  Then there is a *information*() method within each class to print a description about each of the individual employee types. Here all the attributes of the employee will be shown.  Finally in the *main*() class a new employee instance is created with ‘new’ for each employee created in the array. Where the information if outputted to the user as the *printName*() and *draw*() function is called for each employee.  Overall for this exercise, it was a very similar exercise to the second exercise where we were more free to create the classes ourselves and so I took inspiration and basis from exercise two. |

|  |
| --- |
| **Test description**  ***Exercise 1:***  Test 1: Does the code compile.  Result 1: Yes  Test 2: Does the code print ‘hello world’  Result 2: Yes  ***Exercise 2:***  Test 1: Does the code print out each object created?  Result 1: Yes  Test 2: Is what printed out relevant and show all data supplied for each of the shape object types?  Result 2: Yes  Test 3: Does my extra credit addition of finding shapes area display and calculate correctly?  Result 3: Yes  ***Exercise 3:***  Test 1: Does the main() print out each employee object created?  Result 1: Yes  Test 2: Is what printed out relevant and show all data supplied for each of the employee department types?  Test 2: Yes |
| **Known bugs**  None known. |
| **Possible improvements**  A visual display of the shape may be a possible improvement with a graphical view of the measurements and colours to be shown. As this is a nicer display to the user. |
| **Comments**  No comments to be made about this lab. |
| **Extra credit**  *To exercise 2 with the various shapes*. I also added a method to find the area of circle and rectangle objects. This is then presented to the user in the console by the draw() function when called. |
| **References**  None |