

# **COMP285: Computer Aided Software Development Assignment 2 2019/2020**

This is the second of two assessments for COMP285 which contributes 50% of the final module mark.

## **OBJECTIVE**

**This coursework involves the development and testing of some graphics routines in Java™ using the software tools, Eclipse and JUnit.**

## **Assessment Information**

Assignment number	2 of 2
Weighting	50%
Assignment Circulated date provided to class	<b>9/3/2019</b>
Deadline Day & Date & Time	Friday, May 9, 17:00 2020
Submission Mode	Electronic
Learning outcome assessed	<ol style="list-style-type: none"><li>1. Perform software development tasks using the techniques of Automated Testing, Continuous Integration and Test Driven Programming</li><li>2. Use Ant, JUnit and Eclipse both individually and jointly as tools for Automated Testing, Continuous Integration and Test Driven Programming</li></ol>
Submission necessary in order to satisfy Module requirements	No
Purpose of assessment	To assess the students ability to effectively use software development and testing tools
Marking criteria	See end of document

This is the second of two assessments for COMP285 where each contributes 50% of the final module mark.

The general purpose is assessment of the following learning outcomes:

1. Perform software development tasks using the techniques of Automated Testing, Continuous Integration and Test Driven Programming
2. Use Ant, JUnit and Eclipse both individually and jointly as tools for Automated Testing, Continuous Integration and Test Driven Programming with the *main stress* on JUnit testing and test driven programming with Eclipse.

The goal of this assignment is to complete the testing of a partial implementation of a set of shape classes in the style of *Test Driven Programming* in Java by using Unit Testing with JUnit4 in the framework of Eclipse. You are also expected to use test driven development to add a new class which will represent a regular polygon.

You will need to finish/correct this implementation and testing framework. All tests should succeed and all classes should be functioning well.

Each shape class must implement the interface defined IShape.java, familiarise yourself with this source file. This contain comments explaining the required functionality for each shape.

The following classes have already been implemented but some only partially (there is code to add) and some contain bugs. You should have a Junit test class for each of the tested classes.

Point	Represents a point in 2-D
Shape	(abstract base class for all other shape classes)
Rectangle	Represents a rectangle
Circle	Represents a circle
RegularPolygon	Represents a regular polygon
Main	Entry point for the code, also has a rendering function which enables the shapes to be drawn using Java 2D.

## Presentation of work

The project should be implemented in Eclipse and needs to use Junit to test the functionality of **ALL** the public methods of the classes contained in the package uk.ac.liv.comp285.cw1.shapes.

Please follow the package names given in the source file, do not remove any packages and do not define any new packages.

All the actions (such as creating the project, Java source file, JUnit test case, etc.) should be done by Eclipse wizards.

Please name all the test classes by adding in the word Test postfix to all test class names, so to test the Circle class use a class called CircleTest.

The general advice is to work first on something simple. Then there will be fewer problems since you can then depend on what you have already resolved and well-tested.

You should *comment* appropriately all the code you write. This way you can demonstrate your understanding of what they do. This will also be taken into account when marking the work.

All Java files presented by students should be *compilable* and appropriately *formatted*. *Your whole project directory should be zipped up and submitted.*

This zip file should be submitted electronically by the deadline mentioned above via the *Departmental Coursework Submission System*.

Late submissions policy and Plagiarism/collusion are described in Student's Handbook, <http://www.csc.liv.ac.uk/student/ugpdfhandbook.pdf>, Sections, 5.8-5.10.

### **Marking Schema**

This assignment contributes 50% of the final mark for this module, and will be marked according to the following marking scheme.

You must thoroughly test all the classes given and then fix/debug the code provided. To get full marks all methods need to work properly with a full range of appropriate tests to cover all required test partitions.

Please start with testing and debugging the Point class.

#### **Point**

10% All tests covered

10% All code working and complete

#### **Rectangle**

10% All tests covered

10% All code working and complete

#### **Circle**

10% All tests covered

10% All code working and complete

#### **RegularPolygon**

15% All tests covered

15% All code working and complete

#### **Code quality and commenting**

10%