

**FACULTY OF INFORMATICS**

|  |  |  |  |
| --- | --- | --- | --- |
| **SUBJECT’S INFORMATION:** | | | |
| Subject: | CSCI124 Applied Programming | | |
| Session: | July 2014 | | |
| Programme / Section: | J766SENG (SE) / J766CS53 (MGD) / J766CS42 (DSS) | | |
| Lecturer: | Ms. Siti Hawa | | |
| Coursework Type  *(tick appropriate box)* | ❑ Individual Assignment ❑ Group Assignment ❑ Project  ✓Lab Task ❑ Seminar / Tutorial Paper ❑ Others | | |
| Coursework Title: | **Lab Task 8** | Coursework Percentage: | 2% |
| **ASSESSMENT CRITERIA:** | | | |
| Correctness | All programs should produce the correct result as stated in the specification. | | |
| Coding | Programs should use appropriate control structures and data structures correctly based on what have been covered in the class and stated in the specification. Necessary input validations should be done. | | |
| Readability | Appropriate comments are included. Meaningful identifiers used. Proper indentation and line spacing used. | | |
| Well formatted output | Output should be well formatted with appropriate messages displayed. Numbers are shown with appropriate precision. | | |
| **SUBMISSION:** | | | |
| All completed work should be submitted online through Moodle before or on the due date provided.  **SUBMIT AS EARLY AS POSSIBLE. YOU CAN RE-SUBMIT LATER IF NECESSARY. ONLY THE LATEST SUBMISSION WILL BE MARKED.**  **IF YOU SUBMIT YOUR ASSIGNMENT TWICE, ONE SUBMMISSION BEFORE THE DUE DATE AND ANOTHER AFTER THE DUE DATE, THEN YOU WILL BE PENALIZED FOR LATE SUBMISSON.** | | | |
| DUE DATE: | **WEEK 12** | | |
| **PENALTIES FOR LATE SUBMISSION:** | | | |
| Penalties apply to all late work, except if student academic consideration has been granted. Late submissions will attract a penalty of 25% of the assessment mark per day including the weekend. Work more than (3) days late will be awarded a mark of zero. | | | |
| **PLAGIARISM:** | | | |
| **When you submit an assessment task, you are declaring the following**   1. It is your own work and you did not collaborate with or copy from others. 2. You have read and understand your responsibilities under the University of Wollongong's policy on plagiarism. 3. You have not plagiarised from published work (including the internet). Where you have used the work from others, you have referenced it in the text and provided a reference list at the end ot the assignment.   Plagiarism will not be tolerated. Students are responsible for submitting original work for assessment, without plagiarising or cheating, abiding by the University’s policies on Plagiarism as set out in the University Handbook under University Policy Directory and in Faculty handbooks and subject guides. | | | |

**COURSEWORK SPECIFICATION**

**OBJECTIVES:**

The aim of this lab class is to implement three classes. This will help us understand how classes can be used to model the real world.

**TASK 1:**

Declare and implement a class named Circle. Each Circle object should have radius and pi (declared as a constant) as data members. Include a default constructor that sets the radius to 0.0, a set function that takes as arguments the value for the radius of a circle, a function that returns the radius, a function that calculates and returns the area of the circle, a function that calculates and return the diameter of the circle, and a function that calculates and returns the circumference of the circle.

Place your class declaration in Circle.h and the implementations in Circle.cpp.

Implement a driver program that instantiates two Circle objects. Prompt the user to enter values for both objects and display all their information.

**TASK 2:**

Write a class named RetailItem that holds data about an item in a retail shop. The class should have the following data members:

* Item id
* Description
* Units on hand
* Price per item

Write a non-default constructor (initialization constructor) that accepts arguments for each data member, a function that accepts arguments to set the data member values, a function that will display an item’s information, and a function named sold() that will receive a parameter representing a quantity sold. The function sold() should reduce the units on hand (if available) depending on the quantity sold and display the total price for the sold item. If the units on hand are not sufficient, do not deduct the quantity and display a message indicating it is out of stock.

Use appropriate interface and implementation file for your code. Write a driver program that creates a RetailItem object. Then provide three options to the user – change the item’s information, display the item’s information, or sell the item. To sell the item, prompt the user to enter the quantity.