

**FACULTY OF INFORMATICS**

|  |  |  |  |
| --- | --- | --- | --- |
| **SUBJECT’S INFORMATION:** | | | |
| Subject: | CSCI204 Object and Generic 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 3** | Coursework Percentage: | 1% |
| **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 5** | | |
| **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:**

In this lab task, you will learn more about classes and objects. You will also learn how to write constructors and destructor for your classes.

**TASK 1:**

Write a class called Address.

* An address has a house number, a street, a city, a state, and a postal code.
* Supply two constructors: one default constructor that initializes the address to a value of your choice and a non-default constructor that initializes the address with values passed as parameters.
* Include a display function that prints the address on standard output with the house number and street on one line, the city on the next line, and the postal code together with the state on the third line.

Write a main function that instantiate and displays two Address objects. One that uses the default values, and one for which values are entered by the user.

**TASK 2:**

You operate several hot dog stands distributed throughout town.

* Define a class named HotDogStand that has a member variable for the hot dog stand’s ID number and a member variable for how many hot dogs the stand sold that day.
* Create a constructor that allows a user of the class to initialize both values.
* Create a function named justSold that increments the number of hot dogs the stand has sold by one. This function will be invoked each time the stand sells a hot dog so that you can track the total number of hot dogs sold by the stand.
* Add another function that returns the number of hot dogs sold.
* Finally, add a static variable that tracks the total number of hot dogs sold by all hot dog stands and a static function that returns the value in this variable.

Write a main function to test your class with at least three hot dog stands that each sells a variety of hot dogs.

**TASK 3:**

Write a class definition for the class PhoneCall.

* The class should hold a phone number being called, the call length in minutes, and the rate charge per minute.
* Write overloaded constructors – one default constructor, one non-default constructors with initial values that sets the call length and rate charge to 0, and one copy constructor.
* Write a destructor to display a message saying that the object is destroyed.
* Include also necessary accessor and mutator functions.

Write a main function that instantiate 3 PhoneCall objects using different constructor. If a phone call has already been made to a phone number, do not allow the second call to that number. Then display total charges for every phone call made and indicate which phone number that was called with the highest charge.