

**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 8** | 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 14** | | |
| **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 experience how to write programs using exceptions. You are also exposed to drawing use case diagram and class diagram with associations and multiplicities.

**TASK 1:**

You are required to use Singleton design pattern to create a simple Logger application. Each time the logger object will allow only one instance to be created to write the logging messages (code, messages and level (e.g. 1, 2, 3 and 4) into a text file. Incorporate the exception handling concept if users try to create more than one logger object (e.g. Unable to initialize the logger). Users are required to stop/destroy the existing logger instance before start another new process in order to synchronize the file content consistently. Create some useful methods such as log(int code, string message, int level), stopLogger() and etc.

**TASK 2:**

Write a code Symbolic.cpp that contains a function template to display a value preceded and followed by n elements of a symbol x on a line. Write a main() function that tests the function with char, int, double, and string arguments.

The output could be, for example:

\*\*\*47\*\*\*

00039.25000

aaaaBobaaaa

**TASK 3:**

Complete the following tasks:

* Create a calcDistance() function template that accepts two parameters representing two distances from a give point. The function returns the total distance as an integer.
* Create a City class with fields for the city name and for the distance from Penang, the hub city for Air Asia Airlines. Overload the + operator to sum the distance to produce a City result in which the city name contains the two operand city names (for example, “Penang to Kuala Lumpur”) and the distance contains the total distance.
* Create a Planet class with fields for the planet name and distance from Earth. Overload the + operator to sum distances to produce a Planet result in which the planet name contains the two operand planet names (for example, “Venus to Mars”) and the distance contains the total distance.

Write a main() function that declares several integer, double, and City and Planet objects, and uses the calcDistance() function to compute the distance for several pairs.