

**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 6** | 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 11** | | |
| **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 classes, inheritance, and polymorphism.

**TASK 1:**

Design a Ship class that has the following members:

* A member variable for the name of the ship (a string).
* A member variable for the year that the ship was built (a string).
* A constructor an appropriate and appropriate accessors and mutators.
* A virtual print function that displays the ship’s name and the year it was built.

Design a CruiseShip class that is derived from the Ship class. The CruiseShip class should have the following members:

* A member variable for the maximum number of passengers.
* A constructor an appropriate and appropriate accessors and mutators.
* A print function that overrides the print function in the base class. Then function should display only the ship’s name and the maximum number of passengers.

Design a CargoShip class that is derived from the Ship class. The CargoShip class should have the following members:

* A member variable for the cargo capacity in tonnage.
* A constructor an appropriate and appropriate accessors and mutators.
* A print function that overrides the print function in the base class. Then function should display only the ship’s name and the ship’s cargo capacity.

Demonstrate the classes in a program that has an array of Ship pointers. The array elements should be initialised with the addresses of dynamically allocated Ship, CruiseShip, and CargoShip objects. The program should then step through the array, calling each object’s print function.

**TASK 2:**

Code the following in a file AbTop.cpp.

Define classes A, B, and C, such that B is a child of A and C is a child of B. A should be abstract. Each should add a string variable to hold the name of a food beginning with the same letter as the class. Thus, for example, B has a variable BFood and a variable AFood, the latter through inheritance. You should check that appropriate foods do begin with the correct letter.

There should be an output() function which is overridden through the hierarchy. It should provide appropriate information about the objects. Classes B and C should have appropriate constructors, while A needs a function to set its AFood field.

The main() function should construct 2 instances each of B and C. These should be stored in an array of type A. A loop should be used to output information on all the objects, producing something like:

---------------------

A is for Avocado

B is for Burger

---------------------

A is for Acorn

B is for Bean

---------------------

A is for Almond

B is for Blueberry

C is for Cheese

---------------------

A is for Apple

B is for Banana

C is for Cherry

---------------------

**TASK 3:**

Write a program in C++ to demonstrate the use of typeid function. Your program should demonstrate at least five different data types (use a mixture of primitive C++ types and programmer-defined types). Your demonstration may include comparison of different types and display the name of a particular type.