

**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 5** | 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 10** | | |
| **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 and inheritance.

**TASK 1:**

Imagine a publishing company that markets both book and disk versions of its works. Create a class Publication that stores the title (a string) and price (type float) of a publication.

* Derive a class called book from Publication, which adds a page count (type int).
* Add a Disk class that, like book, is derived from Publication. The data item unique to this class is the disk type: either CD or DVD. You can use an enum type to store this item. The user could select the appropriate type by typing c or d.
* Each of these three classes should have a default constructor, a getData() function to get its data from standard input, and a showData() function to display its data.

Write a main() program to test the book and Disk classes by creating instances of them, asking the user to fill in data with getData(), and then displaying the data with showdata().

**TASK 2:**

Define a class named Payment that contains a member variable of type float that stores the amount of the payment, a default constructor, a non-default constructor, a setField() function that receives parameters to set the data member, and a member function named paymentDetails() that outputs the amount of the payment.

Next, define a class named CashPayment that is derived from Payment. Include balance due as the data member for this class. This class should override the paymentDetails() function to add a message to indicate that the payment is in cash and display the balance due if any. Include also a default constructor, a non-default constructor and overload the setFields() function.

Define a class named CreditCardPayment that is also derived from Payment. This class should contain member variables for the name on the card, expiration date, and credit card number. Include a default constructor, a non-default constructor, overload the setFields() functions, and finally, override the paymentDetails() function to include all credit card information in the printout apart from the payment amount.

Create a main function that creates two CashPayment and two CreditCardPayment objects with different values. Show that some objects are initialized using the non-default constructor and some are initialized using the setFields() functions. Display the payment details for each of the objects.

**TASK 3:**

Write code HouseInvest.cpp. There should be two base classes: House and Investment.

The Investment class should contain fields to hold the initial value of an investment, the current value, the profit (different between the previous two fields), and the percentage profit (profit divided by initial value). It should have a constructor that requires initial and current values, and a display function.

The House class should have fields for street address and size (in square meters). It should also have a constructor that sets both fields, and a display function.

Create also a HouseInvest class which inherits from both Investment and House. It includes a constructor, and a display function which uses the parents display functions.

Write a main() function that declares a HouseInvest and displays its values.