

**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 7** | 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 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:**

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: Handling Exceptions**

Write a program that prompts the user to enter a person’s date of birth in numeric form such as 8-27-1980. The program then outputs the date of birth in the form: August 27, 1980. Your program must contain at least two exception classes: invalidDay and invalidMonth. If the user enters an invalid value for day, then the program should throw and catch an invalidDay object. Similar conventions for the invalid values of month and year.

**TASK 2: A Use Case Diagram**

Consider buying petrol from a petrol pump with an attached electronic payment system.

1. Prepare a use case diagram. We assume that a customer will normally pay cash for buying petrol.
2. Add relationships to handle the distinct behaviors of paying by credit card outside, at the pump, or inside, at the cashier.
3. Add another relationship to represent the optional purchase of a car wash.
4. List the actors and explain the relevance or each.
5. Summarize the purpose of each of the four use cases with a sentence.

**TASK 3: Class Relationships**

Consider the following classes.

Company

name

companyID

CarLoan

vehicleID

customerType

customerID

accountNumber

bankID

interestRate

currentBalance

Car

ownerID

vehicleID

ownerType

model

year

Person

name

birthdate

employer1ID

employer2ID

personID

address

Bank

name

bankID

Several of the classes have attributes that are really references to other classes. Such references could be replaced with associations.

* A person may have up to two companies as employers, although this limitation is a little artificial.
* Each person has an ID.
* A car is assigned an ID.
* Cars may be owned by persons, companies, or banks.
* Owner ID refers to the ID of the (single) person, company or bank who/which owns the car.
* Multiple car loans, tied to a single bank, may be involved in the purchase of a car.

Hiding object references as references is the incorrect way to construct a model. You need to prepare a class diagram with using associations and generalizations instead of using IDs. You should try to assign multiplicities. You may need to add a class or two of your own.