1. **COURSEWORK TITLE**

ORDER MANAGEMENT SYSTEM

1. **THE COURSEWORK OVERVIEW & DESCRIPTION**

It’s a very common practice in retail shops to manage the customer orders placed by its various customers. Basically it helps to streamline the important entities involved in a retail shop namely customers, orders and products. You are required to develop an application that simulates a simple retail order management system which involves the real world objects namely customers, orders and products. The users should be able to add customers and products. The developed system should also allow the customer to place orders. Each order has many items, which are the products available in the retail shop. In addition a supporting document is needed to reflect the design of the implementation codes and the implementation details that utilises the Object-oriented programming concepts.

The developed system should be complete with options for adding customers, products and orders with multiple items. You as an Object-oriented programming student need to identify the relationship among the entities and develop the necessary methods needed to fulfil the requirements of the expected systems.

The following are the basic requirements.

**Login access:**

You program should have two types of access rights such as Admin and Customer.

**Admin:**

Admin should have access to the following functionalities:

* Customer (Add/Delete/Edit/View/Search)
* Product (Add/Delete/Edit/View/Search)
* Order (Add/Delete/Edit/View/Search)

**Customer:**

Customer should have access to the following functionalities:

* Order (Add/Delete/Edit/View/Search)

The entities involved in the system is given as follows:

**Order**

* Identify the various attributes needed for describing the order entity.
  + Each item ordered should be a product that exist
  + Identify and include the necessary methods.
  + Include the functionality to calculate the total order amount, which sums up the individual items of that specific order by a particular customer.
  + No duplication of orders allowed.
  + Each order includes many order items.

**OrderItem**

* Identify the various attributes needed for describing the order item entity.
* Order Item holds the details of items placed in a specific order
* Identify and include the necessary methods and each order item should be a product entity

**Customer**

* Identify the various attributes needed for describing the customer entity.
* Identify and include the necessary methods required to make the customer class fully functional.
* No duplication of customers allowed.
* Added customers should be able to place orders.

**Product**

* Identify the various attributes needed for describing the Product entity.
* Product can be of two types: Fragile and Non Fragile
* Identify and include the necessary methods required to make the product class fully functional.
* No duplication of products allowed.
* For every product a bundling/package charge need to be added in the cost. The cost for packing fragile items is more than packing non-fragile items.

1. **OBJECTIVES OF THIS COURSEWORK**

Develop the practical ability to describe, justify, and implement a C++ based system.

1. **LEARNING OUTCOMES**

At the end of this coursework, you should be able to:

* Construct an appropriate C++ solution for a particular scenario (C6,PLO2)
* Demonstrate the ability to propose the solution for a new or existing problem using C++ Programming (A3, PLO6)

1. **TYPE**

Individual Assignment

1. **GENERAL REQUIREMENTS**

* The program submitted should compile and be executed without errors.
* Validation should be done for each entry from the users to avoid logical errors.
* The implementation code **must highlight** the use of object-oriented programming concepts as required by the solution.
* Students should use **text files** for storing and retrieving data required for the system.
* **Not allowed** to use any database tools like access / oracle etc.

1. **DELIVERABLES:**

* The system with complete code to be submitted in the Moodle.
* Report document in softcopy form to be submitted in the Moodle.
* Submission deadline:

1. **DOCUMENTS: COURSEWORK REPORT**

* As part of the assessment, you must submit the project report in softcopy form, which should have the following format:

1. Cover Page:

All reports must be prepared with a *front cover*. A protective transparent plastic sheet can be placed in front of the report to protect the front cover. The front cover should be presented with the following details:

* Module
* Coursework Title
* Intake
* Group member (Student name and ID)
* Date Assigned (the date the report was handed out).
* Date Completed (the date the report is due to be handed in).

1. Contents:

* Description and justification of the design and the implementation code which illustrate the object oriented programming concepts incorporated into the solution
* A 2000-word report based on the object-oriented topic researched

1. Conclusion
2. References

* The font size used in the report must be 12pt and the font is Times New Roman. Full source code is not allowed to be included in the report. The report must be typed and clearly printed.
* You may source algorithms and information from the Internet or books. Proper referencing of the resources should be evident in the document.
* All references must be made using the Harvard Naming Convention as shown below:

*The theory was first propounded in 1970 (Larsen, A.E. 1971), but since then has been refuted; M.K. Larsen (1983) is among those most energetic in their opposition……….*

*/\*\**

*\* Following source code obtained from (Danang, S.N. 2002)*

*\*/*

*int noshape=2;*

*noshape=GetShape();*

* List of references at the end of your document or source code must be specified in the following format:

*Larsen, A.E. 1971, A Guide to the Aquatic Science Literature, McGraw-Hill, London.*

*Larsen, M.K. 1983, British Medical Journal [Online], Available from http://libinfor.ume.maine.edu/acquatic.htm (Accessed 19 November 1995)*

*Danang, S.N., 2002, Finding Similar Images [Online], The Code Project, \*Available from http://www.codeproject.com/bitmap/cbir.asp, [Accessed 14th \*September 2006]*

* Further information on other types of citation is available in *Petrie, A., 2003, UWE Library Services Study Skills: How to reference [online], England, University of Western England, Available from http://www.uweac.uk/library/resources/general/info\_study\_skills/harvard2.htm, [Accessed 4th September 2003].*

1. **ASSIGNMENT ASSESSMENT CRITERIA**

The assignment assessment consists of four components: Requirement Analysis (20%), Implementation (40%), Report (30%), and Presentation (10%). Details of the allocation for each component are as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Program Solution and Design (60%)** | | **Documentation (20%)** | | | |
| **Criteria** | **Marks Allocated** | **Criteria** | **Marks Allocated** | |
| Program Solution : Performance result of all operations and design as per requirements | 40% | Structure of the report and references Content:   * Description and justification of the C++ concepts incorporate * Program out screenshots   Project description, limitation and conclusion | 20% | |
| Program Design: Appropriate design and the implementation of codes illustrating the C++ programming concepts incorporated | 20% |
| **Presentation (20%)** | | | | | |
| **Criteria** | | | | **Marks Allocated** | |
| * Ability to answer questions addressed by the lecturer pertaining to the work done and presented | | | | 20% | |

1. **DEVELOPMENT TOOLS**

The program must be written in C++ language and you can use any development IDE as a tool but the back-end data store must be **.txt** files.

1. **ACADEMIC INTEGRITY**

* You are expected to maintain the utmost level of academic integrity during the duration of the course.
* Plagiarism is a serious offence and will be dealt with according to APU University regulations on plagiarism.