ITECH7201 Software Engineering: Analysis and Design

Assignment 1

Overview

You are required to design and develop a small Java console application. Completion of this assignment requires an understanding of:

- Analysis and design techniques, including development of use cases and UML diagrams – specifically, use case diagrams, class diagrams and sequence diagrams
- Object-oriented programming, focusing on polymorphism and the use of interfaces

Timelines and Expectations

Percentage Value of Task: 20%

Due: 11:55pm Sunday 16th September, 2018 (week 7)

Minimum time expectation: 20 hours

Learning Outcomes Assessed

The following course learning outcomes are assessed by completing this assessment:

- Understand the significance of detailed project planning and control, good communication and documentation and the use of appropriate tools in order to provide a quality product
- Understand the distinction between software engineering and programming, and thus the distinction between a software configuration and a program
- Understand the methods and techniques involved in designing, implementing and maintaining an information system, in particular using an object-oriented approach
- Demonstrate skills in designing and implementing an information system.

Assessment Details

A new retail store is opening in Australia, selling any products of your choice. This company operates a loyalty scheme, whereby members earn points for purchases that can subsequently be used for in-store discounts. Every 1000 points accrued provides a $10 discount. Points are earned at varying rates depending on the customer's membership level:

Membership Level	Points Accrued Per Dollar Spent
Non-Member	0
Standard Member	1
Gold Member	2
Employee	5

You have agreed to design and develop a small Java console program for this company, with four menu options.

Option 1: Create a member. The member needs to have a name, a membership level and a starting point balance of zero.

Option 2: Make a purchase. Select either a guest (non-member) or a member who is making a purchase, show the current point balance, enter the total recommended retail value of that purchase and calculate and apply the new points balance. If the current point balance is 1000 or more at the start of the purchase, the system should deduct $10 from the purchase price entered and calculate the loyalty points based on the discounted purchase price.

Option 3: Display a summary of transaction data for the current session, showing:

- The total number of sales processed and the payment amount received for these sales.
- The total number of sales processed for each membership level, the total loyalty points accrued for each membership level, and the total discounts given at each membership tier.

Option 4: Exit the system

After completing any of the first three options, the program returns to the menu so the user can select another option. After selecting the fourth option, the program closes. There is no need for the program data to persist once the program has closed.

The retail store wants to be able to add additional membership levels at a later date, so the system needs to be flexible. This means you will need to use an interface for processing payments, and polymorphism for the various membership classes, so that new, different levels of membership may be added at a later date with minimal updates to the code. The company has also asked that you provide them with some documentation before you commence coding, so that they are able to verify that the program you intend to code will address their requirements. They would like to see use cases to summarize the requirements in written format, as well as use case diagrams, class diagrams and sequence diagrams.

Submission

You are required to submit the assignment before the due date consisting of:

- A Zip file containing the following (submitted via Moodle under the Assignment 1 link)
  - A written report comprising:
    - Use Cases summarizing the requirements
    - UML Diagrams, created in Enterprise Architect, comprising:
      - a Use Case Diagram for processing a hire agreement
      - a Class Diagram of the intended system
      - a Sequence Diagram for processing a hire agreement for a member.
    - A short reflection (approximately 200-300 words) discussing the importance of requirements design and analysis, UML diagrams and object-oriented programming with interfaces and polymorphism. As an example, if you found that you changed your initial UML diagrams after you had commenced coding, you should explain what these changes were and explain what you learnt that led to these changes.
  - Enterprise Architect file(s) containing your UML Diagrams for the Use Case, Class and Sequence Diagrams
  - Your finished Java program, addressing the requirements outlined in the Assignment Details.

Marking Criteria/Rubric

Task	Available Marks	Student Mark
Requirements Analysis and Design		
• Use Cases summarizing the requirements of the program	2	
• A Use Case Diagram for processing a purchase by a member	2	
• A Class Diagram of the intended system	2	
• A Sequence Diagram for processing a member's purchase	2	
Development of Code		
A complete Java program addressing the requirements outlined in the Assignment Details section of this specification, including:		
• Functionality to create members at each of the membership tiers (including a guest membership account for non-members)	1	
• Functionality to process purchases for each of the membership levels, including display of the correct loyalty point balance and points accrued in the current transaction for that member	2	
• A progressive payments menu option that displays the total number of sales processed and the payment amount received for these sales, the total number of sales processed for each membership level, the total points accrued for each level and the total discounts given at each membership level.	1	
• Code demonstrating the use of an interface and polymorphism to handle loyalty points and the various membership options available	6	
Reflection on Learning		
• A short reflection (approximately 200-300 words) discussing the importance of requirements design and analysis, UML diagrams and object-oriented programming with interfaces and polymorphism.	2	
Total	20	

Feedback

Marks will be uploaded in fdlGrades and a completed marking guide provided in Moodle within 2 weeks of assignment submission.

Plagiarism:

Plagiarism is the presentation of the expressed thought or work of another person as though it is one's own without properly acknowledging that person. You must not allow other students to copy your work and must take care to safeguard against this happening. More information about the plagiarism policy and procedure for the university can be found at http://federation.edu.au/students/learning-and-study/online-help-with/plagiarism.