



Tunku Abdul Rahman University of Management & Technology

AACS2204

Object-Oriented Programming Techniques

Assignment

2023/2024

Programme : Diploma in Computer Science

Tutorial Group : 8

Date Submitted to Tutor : 28 September 2023

Team Members:

No	Student Name	Student ID
1.	Aloysius Khoo	22WMD08707
2.	Gregory Chia Ming Feng	22WMD08692
3.	Khoo Li Xuan	22WMD08619
4.	Nicholas Lim Sze Whye	22WMD08641

No.	Team Member	Task(s) Allocated
1.	Aloysius Khoo	Module: Game menu, add cart module, add game module Order module Classes: Game, Cart, Order
2.	Gregory Chia Ming Feng	Modules: Top Up, Add Bank, Payment Classes: AccountWallet, Credit, Payment
3.	Khoo Li Xuan	Register Module, Login Module Classes : User, Customer, Staff
4.	Nicholas Lim Sze Whye	Staff and Customer Main Menu, Summary Report

Coursework Declaration

We confirm that we have read and shall comply with all the terms and conditions of TAR University College's plagiarism policy.

We declare that this assignment is free from all forms of plagiarism and for all intents and purposes is our own properly derived work.

Signature :				
Name :	Aloysius Khoo	Gregory Chia Ming Feng	Khoo Li Xuan	Nicholas Lim Sze Whye
Date :	27/9/2023	27/9/2023	27/9/2023	27/9/2023

AACS2204 Object-Oriented Programming Technique – Assignment Feedback Form

	<u>Student Names</u>	<u>Total Marks</u>	<u>Contribution</u>	<u>Final Marks</u>
A	Aloysius Khoo		25%	
B	Gregory Chia Ming Feng		25%	
C	Khoo Li Xuan		25%	
D	Nicholas Lim Sze Whye		25%	

Programme of Study: (Diploma in Computer Science)

Tutorial Group: (8)

Note:

- The maximum differences after contribution shall be capped at 10m. Refer to assignment specification for more detail.

CLO 2	Demonstrate an object-oriented program using appropriate programming fundamentals with regards to arrays, methods and exception handling. (P4, PLO3)								
Section A: Program Output & Correctness (25 marks) – Individual Marks									
Criteria	Weight	Max Marks	Excellent	Good	Fair	Poor	Very Poor	Marks (Point x Weight)	Remarks
			5	4	3	2	1		
Completeness of the program	2.0	10m	Completed 100% of the functional requirements. All operations are implemented correctly.	Completed all of the functional requirement but implemented with minor errors.	Completed most of the functional requirement but implemented with errors.	Completed < 50% of the functional requirement and < 50% of the operations implemented correctly	Did not complete any of the functional requirement at all	<u>Student Marks</u> A B C D	
Exception Handling	1.0	5m	Output is correct with comprehensive exception handling.	Output is correct with appropriate exception handling.	Output is correct with little exception handling.	Output is somewhat correct with little proper exception handling	Output is totally wrong with no proper exception handling.	<u>Student Marks</u> A B C D	

Design of the Output	1.0	5m	Excellent formatted output. Program displays more than expected	Good formatted output and program displays as expected.	Appropriate output design and display as expected.	Poorly designed output.	Minimum to no formatted output with messy display.	<u>Student Marks</u> A B C D	
Program Code Quality & Standard	1.0	5m	All identifiers strictly conform to the standard Java naming convention and all are meaningful	Most of the identifiers conform to the standard Java naming convention and are meaningful	Some of the identifiers conform to the standard Java naming convention and some are meaningful	Limited numbers of the identifiers conform to the standard Java naming convention or barely meaningful	All of the identifiers conform to the standard Java naming convention or not meaningful	<u>Student Marks</u> A B C D	
Total Marks (25 marks):							A		
							B		
							C		
							D		

CLO 3	Analyse the concepts of encapsulation, inheritance and polymorphism based on programming problems. (C4, PLO2)								
Section B: Object-Oriented Concept (45 marks) – Group Marks									
Criteria	Weight	Max Marks	Excellent	Good	Fair	Poor	Very Poor	Marks (Point x Weight)	Remarks
			5	4	3	2	1		
Class Implementation	1.0	5m	Identified the proper classes, methods and attributes to solve particular problem.	Identified appropriate classes, methods and attributes to solve particular problem with minor errors.	Able to identify only some classes, methods and attributes to solve particular problem.	Inappropriate classes, methods and attributes are used.	Unable to show understanding on the usage of classes, methods and attributes.		

Object Implementation	1.0	5m	Able to create a structure of object collaborating among themselves to carry out task properly.	Able to create a structure of object collaborating among themselves to carry out task properly with little minor mistakes	Structure of collaborating object created with some mistakes	Structure of collaborating object created are barely correct.	Wrong structure of object collaboration like one object doing everything itself.		
Abstraction	1.0	5m	Define the class at the proper level of abstraction (abstract classes and methods).	Define the class at the proper level of abstraction with minor mistake	Classes defined at the proper level of abstractions with some mistakes	< 50% of correct abstraction is used	Lack of abstraction (abstract classes and methods).		
Encapsulation	2.0	10 m	Completely correct implementation of encapsulation (private modifier, setter and getter methods).	Correct implementation of encapsulation with only some minor mistakes	Acceptable amount of correct implementation of encapsulation is applied in program	< 50% of correct implementation for encapsulation applied in program	Completely incorrect implementation of encapsulation (private modifier, setter and getter methods)		
Inheritance	2.0	10 m	Completely correct use of inheritance (correct use of extends keyword, super & sub class)	Correct implementation of inheritance with only some minor mistakes	Acceptable amount of correct implementation of inheritance is applied in program	< 50% of correct implementation for inheritance applied in program	Completely incorrect use and implementation of inheritance		
Polymorphism	2.0	10 m	Completely correct use and implementation of polymorphism. Methods toString() & equals() correctly overridden.	Correct implementation of polymorphism with only some minor mistakes	Acceptable amount of correct implementation of polymorphism is applied in program	< 50% of correct implementation for polymorphism applied in program	Completely incorrect use and implementation of polymorphism.		
Total Marks (45 marks):									

CLO 3	Analyse the concepts of encapsulation, inheritance and polymorphism based on programming problems. (C4, PLO2)
--------------	---

Section C: Object-Oriented Design (30 marks) – Group Marks

Criteria	Weight	Max Marks	Excellent	Good	Fair	Poor	Very Poor	Marks (Point x Weight)	Remarks
			5	4	3	2	1		
Cohesion	1.0	5m	The code demonstrates exceptional cohesion with all modules, classes, or functions have a clear and single responsibility.	The code demonstrates good cohesion overall where most modules, classes, or functions have a clear and single responsibility.	Some modules, classes, or functions have a clear and single responsibility, but others may be slightly ambiguous.	Few modules, classes, or functions have a clear and single responsibility.	Modules, classes represent more than one entity and lack of clear responsibility. Poor class cohesion.		
Coupling	1.0	5m	Minimal or no direct dependency between modules, classes, or functions. Coupling is effectively managed, resulting in highly maintainable and modular code.	Most modules, classes, or functions have minimal direct dependencies.	Some modules, classes, or functions have moderate direct dependencies.	Many modules, classes, or functions have significant direct dependencies.	Modules, classes, or functions have excessive and tightly coupled dependencies which make module be harder to reuse or test.		
Association, Aggregation and Composition	2.0	10m	Associations between classes are accurately identified and properly established.	Associations between classes are generally identified and properly established.	Some associations between classes may be missing or improperly established.	Several associations between classes are missing or improperly established.	Completely incorrect use and implementation of association, aggregation and composition relationship.		
UML Class Diagram	2.0	10m	The UML class diagram demonstrates excellent understanding and implementation. All classes, attributes, and methods are accurately represented with appropriate	Most classes, attributes, and methods are accurately represented with appropriate visibility, data types, and associations.	Some classes, attributes, and methods may be missing or inaccurately represented, with visibility, data types, or associations	Many classes, attributes, and methods are missing or inaccurately represented, with visibility, data types, or associations not adequately defined.	Classes, attributes, and methods are missing or inaccurately represented, with visibility, data types, or associations lacking clarity or definition.		

			visibility, data types, and associations.		not fully define d.				
Total Marks (30 marks):									

Table Of Contents

Table Of Contents	7
System Overview	8
System UML Class Diagram	9
System Modules and Functionality	10
1.0 Customer Register - Khoo Li Xuan.....	10
2.0 Login - Khoo Li Xuan.....	12
3.0 Staff Main Menu - Nicholas Lim Sze Whye.....	13
4.0 Customer Main Menu - Nicholas Lim Sze Whye.....	13
5.0 Game Menu - Aloysius Khoo.....	14
6.0 Cart - Aloysius Khoo.....	15
7.0 Add Game - Aloysius Khoo.....	17
8.0 Top-Up - Gregory Chia Ming Feng.....	20
8.1 Opening Top-Up.....	20
8.2 Description.....	21
8.3 Performing Top-Up.....	22
8.4 Exiting Top-Up.....	23
9.0 Payment - Gregory Chia Ming Feng.....	24
9.1 Accessing Payment.....	24
9.2 Description.....	25
9.3 Account Wallet Method.....	25
9.4 Proceeding With Payment.....	26
9.5 Error Checking.....	27
10.0 Credit Card - Gregory Chia Ming Feng.....	27
10.1 Accessing Credit Card.....	27
10.2 Credit Card Payment Method.....	28
11.0 Summary Module - Nicholas Lim Sze Whye.....	30

System Overview

The Snack Falcon is an online video game point of sales system dedicated to selling virtual video game copies to customers. Staff can also login into the system to add or delete games on sale.

After the global pandemic, selling physical game copies have become difficult over the years. Therefore, our team has come up with a solution to stabilize our sales of video games by integrating an online, contactless, way of selling copies of games through virtual means.

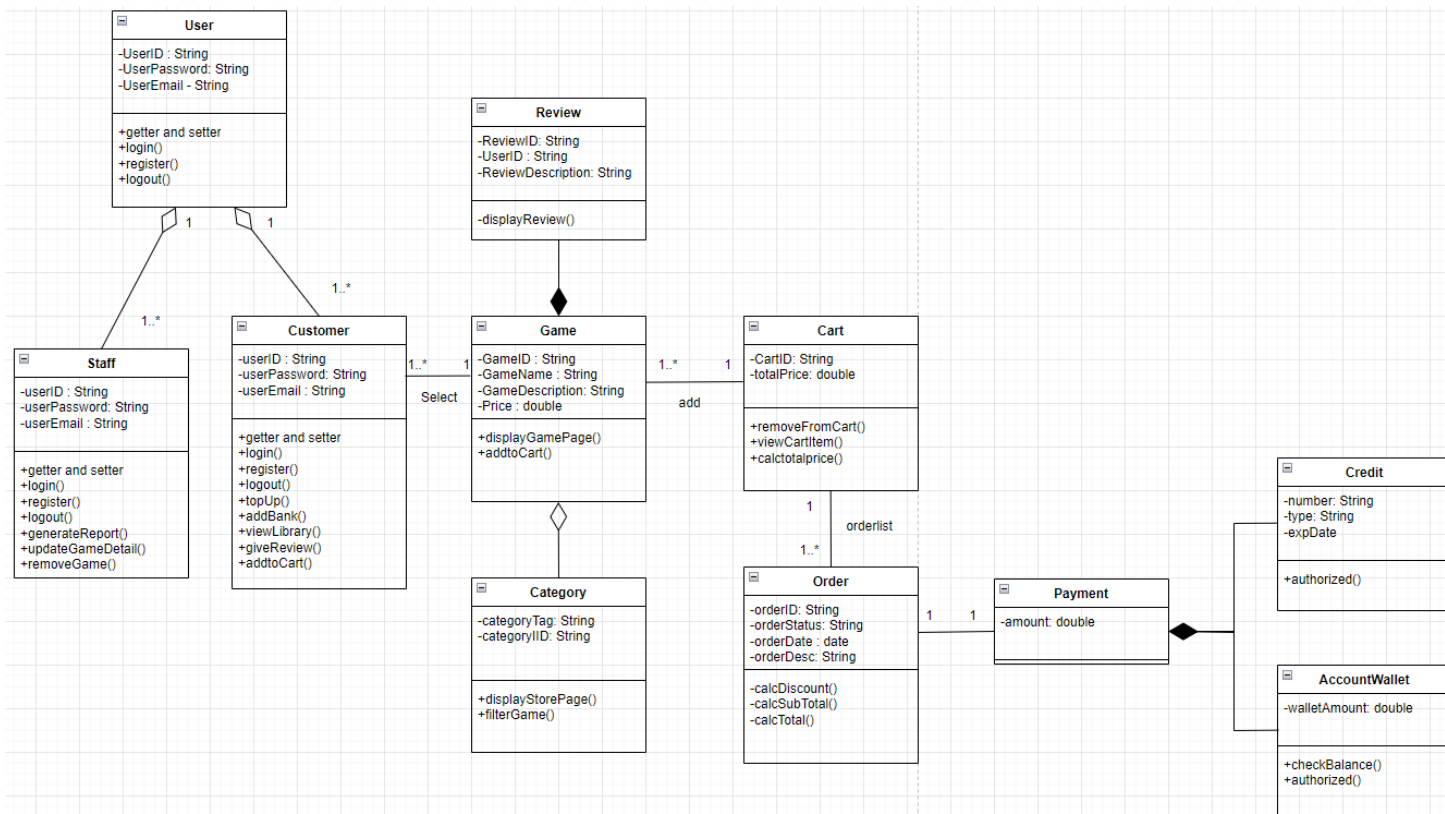
Customers are able to make an account before accessing the system through the login/register module. After signing in, customers are greeted with a main menu. Here, customers can choose between displaying games on sale, viewing carts, adding a credit card, topping up account balance, and reviewing purchased games.

Upon logging in, Staff is able to make modifications to system files through the system. They are able to add new games to the sales system as well as remove them. Staff can also view a summary report on how many games are purchased.

This system is envisioned to assist our growth as a video game distribution company.

System UML Class Diagram

<https://drive.google.com/file/d/1a1NLAJyC73kwwFj4x0Ab5D2BVXYGiyQM/view>



System Modules and Functionality

1.0 Customer Register - Khoo Li Xuan

When the program starts, users will be presented with 3 options, register, login and exit.

```
|<=====>|
|  ===++===  ||          /=====>>          |  //  | | | | | | | | | | |
|    []      ||      //=====  (          |  //  |
|    []      |=====|  |_____  \=====\  //==\ \  //==\ \  //  |
|    []      ||  ||  |          )  ||  ||  \ \_||  [  |  \  |
|    []      ||  ||  L=====  =====/  ||  ||  ||  \ \_||  |  \ \  |
|
|  //==>  ___,  //  _____.  L  _____  R
|  //      /___/  //  //      /==@  L===\ \  //      X  \ \
|  //==  /___/  //  //      /___/  //  //  /  [ ]  Y  A  \
|  //      //  //  \___.  @===/  //  //  \___/  _____  B  /
|                                     \___/  \___/
|<=====>|
```

```
<===== Welcome to The Snack Falcon =====>
```

1. REGISTER
2. LOGIN
3. EXIT

```
Enter choice > |
```

Given that you cannot login if you are not a staff member and do not have an account, customers without an account will have to register themselves at the register function. There the customer id will be automatically generated and assigned to the customer, the system will print out the assigned id to show the customer .

```
Enter choice > 1
[?] Type exit to return to title screen

New ID > C1007
Enter Password >
```

The customer will then be prompted to enter a password and their email. After entering the relevant information, the customer will then be re-shown their assigned id and information they entered, they will then be prompted if they wish to proceed and confirm their registration.

```

Enter Email > sevenOfNine@mail.com
ID      : C1007
Password : pass7
Email    : sevenOfNine@mail.com

```

```

Confirm ? (Y = Proceed / N = Return to Title Screen) >

```

If the customer presses “y”, their relevant data will be stored and they will be brought back to the title screen,

```

Enter Email > guhh@mail.com
ID      : C1008
Password : guhh
Email    : guhh@mail.com

```

```

Confirm ? (Y = Proceed / N = Return to Title Screen) >

```

```

y

```

```

Register succesful!

```

```

|<=====>|
|  ===++=== ||                /=====>>                |  //  | | | | | | | | | |
|      []    ||      //===== (                |  //  |
|      []    |====| |_____\===== \ //==\ \ //==\ \ //  |  |  |
|      []    ||  || |                ) ||  || \ \__||  [  |  \  |
|      []    ||  || L=====      =====/  ||  ||  ||  \ \__  |  \ \  |
|
|      //====>  ___,  //  _____.                L  _____  R  |
|      //      /  /  //  //      /====@  L====\ \      //      X  \ \  |
|      //====  /___/  //  //      /  /  //  //      /  [ ]  Y  A  \  |
|      //      //  //  \___.  @====/  //  //      \  _____  B  /  |
|
|<=====>|

```

```

<===== Welcome to The Snack Falcon =====>

```

1. REGISTER
2. LOGIN
3. EXIT

if the customer presses “n”, their relevant data will not be stored and they will be brought back to the title screen.

2.0 Login - Khoo Li Xuan

When the program starts, users will be presented with 3 options, register, login and exit. Selecting option 2 will then prompt the user to enter their id, when a correct, existing id is entered, users will then be prompted to enter their passwords. If the password is correct, and depending on the type of user id entered, staff or customer id, will then bring the user to their designated menus. Staff will be brought to the staff menu and customers will be brought to the customer menu.

```
Enter ID      > S1000
Login in as Staff...
ID found! Staff ID : S1000
```

```
Enter choice > 2
Type 'exit' to return to title screen
```

```
Enter ID      > C1000
Logging in...
ID found! Customer ID : C1000
```

```
Enter Password > pass0
Login Success!
```

```
=====
      (===M===)  _____  []
      |   |   |  [_____]  ____  [===
      o   o   o  [_____]  ||   o   o
      x       x  |   |   ||   x   x
=====
```

Welcome to Main Menu!

1. Games on Sale
2. Open Cart
3. Top-Up Wallet

If the word “exit” is entered at any time from when they are prompted to enter their id or password, they will instead be brought back to the title screen.

Login Success!

```
=====
      (===M===)  _____  []
      |      |      |  [_____|  ____  [===
      O      o      O  [      |  ||   O    O
      X          X  |      |  ||   X    X
=====
```

Welcome to Main Menu!

1. Games on Sale
2. Open Cart
3. Top-Up Wallet

0. Exit Program

Choice > 0

See you next time!

5.0 Game Menu - Aloysius Khoo

If the Customer select the game menu option, they can access to the game menu page and look at what they want to buy, pressing one of the game will display more detail like their description , price and genre.

```

=====
X====  _ _ _ _ _  |==_==| _ _ _ _ _
| _ _  | _ | | | | _ _  | | | | _ _ | | _ |
| _ _  | | | | | | _ _  | | | | _ _ | | | |
=====

1) Night in the Woods
2) Undertale
3) Hollow Knight
4) ELDEN RING
5) Detroit: Become Human
6) Baldur's Gate 3
7) Rainbow Six Siege
8) Cyberpunk 2077
9) Remnant 2
10) Resident Evil Village
11) Gregs ATD
12) Phasmophobia

0) Exit Game Menu
Select a game >

```

6.0 Cart - Aloysius Khoo

When a Customer want to buy a game they can add it into their cart and continue browsing until they are satisfied with what they have and then they will proceed to Order page and finally Payment

Select a game / 1

```
=====
Game Name   : Night in the Woods
Game Price  : 49.00
Game Genre  : Adventure
=====
```

```
-----
Game Description
-----
```

```
College dropout Mae Borowski returns
home to the crumbling former mining town
of Possum Springs Seeking to resume her
aimless former life and reconnect with
the friends she left behind. But things
are not the same. Home seems different
now and her friends have grown and
changed. Leaves are falling and the wind
is growing colder. Strange things are
happening as the light fades. And there
is something in the woods.
```

```
=====
[1] Add to Cart      [2] Back to Games
Please Enter An Option (1-2):||
```

```
=====
(==M==)  _____  []
|  |  |  [__|  _  [==
O  o  O  [  |  ||  O  O
X      X  |  |  ||  X  X
```

```
=====
Welcome to Main Menu!
```

1. Games on Sale
2. Open Cart
3. Top-Up Wallet

0. Exit Program

Choice > 2

[Your Cart is Empty!!]


```

=====
[1] Add to Cart      [2] Back to Games
Please Enter An Option (1-2):1

===== Your Cart Content =====
Game Name                                     Price

Undertale                                     23.00
ELDEN RING                                    199.00
-----

Total price:                                222.00

Continue Looking For Games? (Y/N) >
|

```

```

===== Your Order =====
Order number  > 01002
Order Date    > Thu Sep 28 23:35:47 MYT 2023
Order Desc    >
Subtotal      > 222.00
Tax Rate(5%)  > 11.10
Total         > 233.10

Your total price is... 233.10!

```

7.0 Add Game - Aloysius Khoo

Staff have a function allows them to add a new game on the store by entering these properties, (game name, price, genre, and the description of it), After one game is added, it will automatically generate the GameID for the newly added game.

```

Enter ID      > S1000
Loggin in as Staff...
ID found! Staff ID : S1000

Enter Password > staff1
Login Success!

Please select a Staff Option?
1. Add Game
2. View Report
3. Exit Program

```

Staff have to login first in gain access to the option to add game

```

Enter Password > staff1
Login Success!
    Please select a Staff Option?
    1. Add Game
    2. View Report
    3. Exit Program

1
Add game comfirm? (Yes = Y) (No = N) >
a
Please only select Yes [Y] or No [N]
3
Please only select Yes [Y] or No [N]
|

```

Validation to confirm whether staff really want to proceed with this function, only allow to press yes (Y) or no(N)

```

1
Add game comfirm? (Yes = Y) (No = N) >
a
Please only select Yes [Y] or No [N]
3
Please only select Yes [Y] or No [N]
y

New Game Name: Dead Space

New Game Price: 199
New Game Genre (Enter the number for a Genre)
-----
1)RPG           5)Adventure
2)Action        6)Horror
3)Shooter       7)Relaxing
4)Story Rich    8)Strategy
6

New Game Description: The sci-fi survival-horror classic returns, completely rebu

```

Now after inserting the properties to add a new game, the text file now will have the new game in and when Customer open the program again they will see the new game.

G1000|Night in the Woods|49.00|Adventure|College dropout Mae Borowski returns home to the crumbling former mining town of Possum Springs Seeking
 G1001|Undertale|23.00|Adventure|Welcome to UNDERTALE. In this RPG, you control a human who falls underground into the world of monsters. Now y
 G1002|Hollow Knight|38.50|Action|Forge your own path in Hollow Knight! An epic action adventure through a vast ruined kingdom of insects and h
 G1003|ELDEN RING|199.00|RPG|THE NEW FANTASY ACTION RPG. Rise, Tarnished, and be guided by grace to brandish the power of the Elden Ring and be
 G1004|Detroit: Become Human|107.50|Story Rich|Detroit 2038. Technology has evolved to a point where human like androids are everywhere. They s
 G1005|Baldur's Gate 3|219.00|RPG|Baldur's Gate 3 is a story-rich, party-based RPG set in the universe of Dungeons & Dragons, where your choice
 G1006|Rainbow Six Siege|65.00|Shooter|Tom Clancy's Rainbow Six® Siege is an elite, tactical team-based shooter where superior planning and exe
 G1007|Cyberpunk 2077|219.00|RPG|Cyberpunk 2077 is an open-world, action-adventure RPG set in the dark future of Night City — a dangerous megal
 G1008|Remnant 2|156.00|Shooter|Remnant II pits survivors of humanity against new deadly creatures and god-like bosses across terrifying worlds
 G1009|Resident Evil Village|163.00|Horror|Experience survival horror like never before in the 8th major installment in the Resident Evil franchi
 G1010|Gregs ATDS|10.00|Strategy|A Tower Defense Game made by Gregory from TARUMT. A game featuring an array of animals to add to your arsenal
 G1011|Phasmophobia|31.00|Horror|Phasmophobia is a 4 player online co-op psychological horror. Paranormal activity is on the rise and it's up to
 G1012|Dead Space|199.00|Horror|The sci-fi survival-horror classic returns, completely rebuilt to offer an even more immersive experience — inc

```

=====
X====  _ _ _ _ _ |==_==| _ _ _ _ _
| _ _ | _ | | | | _ _ | | | | _ _ | | | | _ _ |
| _ _ | | | | | | _ _ | | | | _ _ | | | | _ _ |
=====

```

- 1) Night in the Woods
- 2) Undertale
- 3) Hollow Knight
- 4) ELDEN RING
- 5) Detroit: Become Human
- 6) Baldur's Gate 3
- 7) Rainbow Six Siege
- 8) Cyberpunk 2077
- 9) Remnant 2
- 10) Resident Evil Village
- 11) Gregs ATDS
- 12) Phasmophobia
- 13) Dead Space

8.0 Top-Up - Gregory Chia Ming Feng

8.1 Opening Top-Up

```

=====
      (===M===)  _____  []
      |  |  |  [_____]  _  [===
      0  o  0  [  |  ||  0  0
      X      X  |  |  ||  X  X
=====

Welcome to Main Menu!
1. Games on Sale
2. Open Cart
3. Top-Up Wallet
0. Exit Program
Choice > 3
=====

Welcome to Account Wallet Top-Up!
ID      : AW1001
Balance : RM 0.0
=====

Please input amount to Top-Up below:
Do 'X' to Exit

```

From the **Main Menu**, users are able to perform a Top-Up by selecting the Top-Up wallet option (**Choice 3**). The goal of the top-up module is to allow users to increment their balance in their Account Wallets before making a purchase.

```

Your Wallet Balance: 0.0
Total Price          : 49.0
1. Yes, pay with Account Wallet.
2. Cancel, return to Main Menu.

```

```

Choice > 1
Insufficient Funds!!
Please select another option:
1. Top-Up Account Wallet
2. Return to Main Menu
3. Return to Cart Menu
4. Retry Payment Methods

```

Option >

The Top-Up module can also be accessed from **Payment Module (Choice 1)**, if and only if the user has insufficient balance when choosing “Account Wallet” as their payment method.

8.2 Description

```
=====
                Welcome to Account Wallet Top-Up!
ID      : AW1001
Balance : RM 0.0
=====

Please input amount to Top-Up below:
Do 'X' to Exit
.
```

This is the **Top-Up Menu**.

```
=====
                Welcome to Account Wallet Top-Up!
ID      : AW1001
Balance : RM 0.0
=====

Please input amount to Top-Up below:
Do 'X' to Exit
```

Account Wallet ID : When User logs in, a new AccountWallet object is created for that user. This wallet holds the user's balance, which starts with zero. After checkout, the change will be reimbursed to user.

Balance : Top-Up menu displays user's current balance which updates accordingly.

Instructions : Users are prompted to input how much money they want to insert into their wallets. Here, we assume users have handed cash physically or by other means.

8.3 Performing Top-Up

```
=====
                Welcome to Account Wallet Top-Up!
ID       : AW1001
Balance  : RM 0.0
=====

Please input amount to Top-Up below:
Do 'X' to Exit
100
Successful Top-Up!!
Your Current Balance is: 100.0
=====

      (===M===)  _____  []
      |  |  |  [_____]  _  [===
      O  o  O  [  |  ||  O  O
      X      X  |  |  ||  X  X
=====

Welcome to Main Menu!
1. Games on Sale
2. Open Cart
3. Top-Up Wallet

0. Exit Program
Choice > |
```

Let's do a mock test. By inserting a top-up amount of **100**,
The module accepts the value and **updates** the AccountWallet balance.

```
=====
                Welcome to Account Wallet Top-Up!
ID       : AW1001
Balance  : RM 100.0
=====

Please input amount to Top-Up below:
Do 'X' to Exit
69.69
Successful Top-Up!!
Your Current Balance is: 169.69
=====

      (===M===)  _____  []
      |  |  |  [_____]  _  [===
      O  o  O  [  |  ||  O  O
      X      X  |  |  ||  X  X
=====

Welcome to Main Menu!
1. Games on Sale
2. Open Cart
3. Top-Up Wallet

0. Exit Program
Choice >
```

Let's perform another mock test. This time, inputting **decimals**.
AccountWallet balance is **updated** accordingly.

8.4 Exiting Top-Up

```

=====
                Welcome to Account Wallet Top-Up!
ID       : AW1001
Balance  : RM 169.69
=====

Please input amount to Top-Up below:
Do 'X' to Exit
X
=====

      (===M===)  _____  []
      |   |   |  [_____]  _  [===
      O   o   O  [  |   |  |  O   O
      X       X  |   |   |  X   X
=====

Welcome to Main Menu!
1. Games on Sale
2. Open Cart
3. Top-Up Wallet

0. Exit Program
Choice >

```

Inputting 'X' or 'x' allows users to **Quit** the Top-Up Menu and return back to the Main Menu.

9.2 Description

```
=====
0===
|_| | | | | | | |
|_| |_| |_| | | | | | ==|==
|_| |_| |_| | | | | | |
=====
```

```
Select Payment Type:
1. Credit Card
2. Account Wallet

Choice >
```

Here, users are prompted to enter a choice between paying with **Credit Card** or **Account Wallet**.

9.3 Account Wallet Method

```
=====
0===
|_| | | | | | | |
|_| |_| |_| | | | | | ==|==
|_| |_| |_| | | | | | |
=====
```

```
Select Payment Type:
1. Credit Card
2. Account Wallet
```

```
Choice > 2
```

```
Your Wallet Balance: 1000.0
Total Price          : 49.0
1. Yes, pay with Account Wallet.
2. Cancel, return to Main Menu.
```

```
Choice >
```

Choosing to pay using Account Wallet (from Top-Up Module) will have the program display the Account Wallet balance as well as the total price from the Cart.

Here, users can pick to **Proceed** or **Return**.

9.5 Error Checking

```
=====
O====-   _   _   |
|         |__| [   =|==
O====O   |   | [   _

=====

Displaying Cart:

Name                                           Price

Night in the Woods                               49.00

=====

Please pick an option:

1. Proceed with Checkout Order
2. Return to Main Menu

Option > 1

=====

O===
|__| _   _   _   _   |
|   |__| |__| |   |__| |   ==|==
|   |   | _| |   | |__| |   _

=====

Select Payment Type:

1. Credit Card
2. Account Wallet

Choice > awdsa
Only Enter number!
Select Payment Type:

1. Credit Card
2. Account Wallet

Choice >
```

In **Payment Menu**, if an invalid input is inserted, users will be prompted with error message and re-prompted with the choice.

10.0 Credit Card - Gregory Chia Ming Feng

10.1 Accessing Credit Card

```
Select Payment Type:
1. Credit Card
2. Account Wallet

Choice > 1

== Adding Bank Account ==
Do 'X' to Exit To Main Menu
Enter Bank Associated with Account >
```

The Credit Card module can be accessed through the first choice in the payment module, after accessing it you will have the ability to connect your credit card with the system and buy your games without using top-up. Customers are required to enter the bank they are associated with and the bank account number that they have on their CreditCard, after inserting the payment module will proceed with the transaction. After the transaction is completed the Customer will see their receipt and their total spend.

10.2 Credit Card Payment Method

Bank Account details are then required to be entered in order to make payments and after successfully making their payment, it will generate a receipt on the game purchased and the total amount spent.

```
== Adding Bank Account =====
Do 'X' to Exit To Main Menu
```

```
Enter Bank Associated with Account >
Maybank
Enter Bank Account Number (8 digits) >
```

```
== Adding Bank Account =====
Do 'X' to Exit To Main Menu
```

```
Enter Bank Associated with Account >
Maybank
Enter Bank Account Number (8 digits) >
12345678
Enter Card Expiration Date (MM/YY) >
```

```
== Adding Bank Account =====
Do 'X' to Exit To Main Menu
```

```
Enter Bank Associated with Account >
Maybank
Enter Bank Account Number (8 digits) >
12345678
Enter Card Expiration Date (MM/YY) >
12/12
```

```
Successfully Added a Credit Card!
```

```
=====
Account Number > 12345678
Account Type   > Maybank
Expiry Date    > 12/12
=====
```

```
Successful Payment!
```

```
===== RECEIPT =====
```

```
Game Purchase      Price
Night in the Woods  49.00
```

```
-----
Your bank account has been charged.
THANK YOU FOR SHOPPING, BYE BYE~
```

Successful Adding Credit Card: After inputting all credit card information, user is prompted with a success message as well as the Credit Card Information.

Successful Payment with Credit Card: After inputting all credit card information, user is prompted with a success message as well as Receipt.

11.0 Summary Module - Nicholas Lim Sze Whye

After checking out and purchasing the games, only staff are able to generate a summary report that displays the game name, quantity and amount spent on each game.

Welcome to Summary Report!		
Games	Quantity Sold	Amount
Night in the Woods	1	49.00
Undertale	0	0.00
Hollow Knight	0	0.00
ELDEN RING	0	0.00
Detroit: Become Human	0	0.00
Baldur's Gate 3	0	0.00
Rainbow Six Siege	0	0.00
Cyberpunk 2077	0	0.00
Remnant 2	0	0.00
Resident Evil Village	0	0.00
Gregs ATD	0	0.00