

Instructions to candidates

This paper consists of Three (03) parts: Part A (MCQ), Part B (Structured) and Part C (Practical)

Total mark allocation: 100 marks

Answer all the questions.

Part C

Number of questions: Two (02) essay questions

Time allocation: (02) hours

Mark allocation: 50 marks

Create a folder in the desktop (i.e. UWU_CST_21XXX) with two (02) sub folders for each question.

Save your Java source code files inside the relevant sub folder according to the question number.

Zip the main folder and upload it to the correct link given in the VLE course page.

You should frequently save the project.

1. Write a Java program to handle the inventory of a shop considering the following scenario.

- a. Create a class called **Product** with **name**, **unit_price** and **stock_quantity** attributes. Use appropriate access modifiers and data types for defining the variables.
(02 marks)
- b. Implement a parameterized constructor for the **Product** class with three parameters that sets the **name**, **unit_price** and **stock_quantity** to any desired values.
(02 marks)
- c. Create **four (04)** objects from the **Product** class using the parameterized constructor with some desired values for the object attributes.
(04 marks)
- d. Generate an inventory report by displaying the following formatted output using all the objects you created.

Inventory Report :

| Product Name | Unit Price (Rs.) | Available Quantity | Total Price (Rs.) |
|--------------|------------------|--------------------|-------------------|
| Soap | 120.00 | 100 | 12000.00 |
| Cheese | 450.00 | 50 | 22500.00 |
| Book | 250.00 | 150 | 37500.00 |
| Pen | 80.00 | 120 | 9600.00 |
| Ice Cream | 500.00 | 80 | 40000.00 |
| Milk | 240.00 | 160 | 38400.00 |
| Total | | | 160000.00 |

(10 marks)

- e. Create **two (02)** sub classes called **Stationary** and **Foods** those inherit from the **Product** class.

(02 marks)

2. Write a Java program to handle the operations of a bank considering for the following scenario.

- a. Create a class called **OnlineBankingApplication** with private fields for **username**, **password** and **account_balance**. (Use appropriate data types for defining these variables.)

(02 marks)

- b. Initialize desired values for **username**, **password** fields and a default value for the **account_balance** field.

(02 marks)

- c. Implement public getter methods for each field to retrieve bank account information.

(06 marks)

- d. Implement a public setter method for the **account_balance** field to update the bank account balance.

(02 marks)

- e. Create a class called **BankManagement** to authenticate a user. Create a method called **authenticate()** inside this class which takes the **username** and the **password** as user inputs. (Trim the input **username** to remove any leading or ending white spaces.)

(04 marks)

- f. Using the getter methods that you defined in 2(c), verify the **username** and **password** of the user.

(03 marks)

- g. Create a method called **getBalance()** to display the amount of money in the bank account. Use the getter methods that you defined in 2(c) when retrieving the bank account balance.
(02 marks)
- h. Create a method called **deposit()** to deposit money into the bank account. Get the depositing amount as a user input and store that value into the **account_balance** field using the setter method that you defined in 2(d).
(03 marks)
- i. Create a method called **withdraw()** to withdraw money from the bank account. Get the withdrawing amount as a user input and deduct that value from the **account_balance** field using the getter and setter methods that you defined in 2(c) and 2(d).
(03 marks)
- j. Define the main method inside the **BankManagement** class. Call the **authenticate()** method when starting the program. Upon successful authentication of the user, call the necessary operations (getBalance, withdraw, deposit) depending on the user's choice as shown below.

Select an operation :

1. Check Balance
2. Deposit Money
3. Withdraw Money

Enter your choice :

(03 marks)

[End of Paper]