Object Oriented Programming (BCSF20), Morning

Dated: 1-11-2021

20 marks

Note:

- All the programs should be implemented using classes. You can take input in main() function and then call appropriate methods/ member functions of a designed class to set and get values.
- The attributes of class should be declared as private and member functions as public.
- You should not initialize the attributes while declaring them in class. The values should be assigned using member functions only. E.g. you cannot declare like:

```
Class Person
{
        private:
        int age=25;
}
```

- All inputs should be taken in main() and all the final results should also be reported/ displayed in the main function.
- All the logic should be implemented in class' member functions. Main() should only input and output relevant values by calling relevant functions of the class

Last hour is reserved for evaluation purpose. Make sure to complete the tasks in 2 hours time.

Lab 2

Question 1: (10 marks)

Create an Account class that a bank might use to represent customers' bank accounts. Include a data member of type int to represent the account balance. Provide a constructor that receives an initial balance and uses it to initialize the data member. The constructor should validate the initial balance to ensure that it's greater than or equal to 0. If not, set the balance to 0 and display an error message indicating that the initial balance was invalid. Provide three member functions. Member function credit should add an amount to the current balance. Member function debit should withdraw money from the Account and ensure that the debit amount does not exceed the Account's balance. If it does, the balance should be left unchanged and the function should print a message indicating "Debit amount exceeded account balance." Member function getBalance should return the current balance. Create a program that creates two Account objects and tests the member functions of class Account.

Question 2: (10 marks)

Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four data members—a part number (type string), a part description (type string), a quantity of the item being purchased (type int) and a price per item (type int). [Note: In subsequent chapters, we'll use numbers that contain decimal points (e.g., 2.75)—called floating-point values—to represent dollar amounts.] Your class should have a constructor that initializes the four data members. A constructor that receives multiple arguments is defined with the form: ClassName(TypeName1 parameterName1, TypeName2 parameterName2, ...) Provide a set and a get function for each data member. In addition, provide a member function named getInvoiceAmount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as an int value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0. Write a test program that demonstrates class Invoice’s capabilities