University of Central Punjab

Object Oriented Programming

Assignment # 1

Start Date: 05-11-2022 **Section: C3** **Total Marks: 100**

Due Date: 12-11-2022 **Program:** BSCS

Instructions

1. Understanding of the problems is part of the assignments.
2. You will get Zero marks if found any type of cheating.

Q#1: Create an 'Account' class that a bank might use to represent customers' bank accounts. Include a data member of type **float** 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. (Marks 20)

Q#2: 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 **float**) and a price per item (type **float**). Your class should have a constructor that initializes the four data members. 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. (Marks 25)

Q#3: What's a default constructor? How are an object's data members initialized if a class has only an implicitly defined default constructor? (Marks 05)

Q#4: Create a Time Class with all necessary members and provide a constructor that's capable of using the current time from the time and `localtime` functions—declared in the C++ Standard Library header `<ctime>`—to initialize an object of the Time class. (Marks 20)

Q#5: Create a class called Complex for performing arithmetic with complex numbers. Write a program to test your class. Complex numbers have the form

```
realPart + imaginaryPart * i
```

where `i` is $\sqrt{-1}$

Use double variables to represent the private data of the class. Provide a constructor that enables an object of this class to be initialized when it's declared. The constructor should contain default values in case no initializers are provided. Provide public member functions that perform the following tasks:

a) Adding two Complex numbers: The real parts are added together and the imaginary parts are added together.

b) Subtracting two Complex numbers: The real part of the right operand is subtracted from the real part of the left operand, and the imaginary part of the right operand is subtracted from the imaginary part of the left operand.

c) Printing Complex numbers in the form (a, b), where a is the real part and b is the imaginary part. (Marks 20)

Q#6: Create a class called Employee that includes three pieces of information as data members—a first name (type **string**), a last name (type **string**) and a monthly salary (type **float**). Your class should have a constructor that initializes the three data members. Provide a *set* and a *get* function for each data member. If the monthly salary is not positive, set it to 0. Write a test program that demonstrates class Employee's capabilities. Create two Employee objects and display each object's *yearly* salary. Then give each Employee a 10 percent raise and display each Employee's yearly salary again. (Marks 20)