

## **Object Oriented Programing (CL-217)**

### **Lab 04**

**Deadline: Sunday, February 27, 2022 (02:00 PM After noon) (Submit on Google Classroom)**

**Points: 60**

#### **Instructions:**

1. Solve each problem in separate file, Name the code file with problem no (Task\_01, Task\_02,.)
2. Copy these files (Task\_01, Task\_02,.) in a folder and name the folder like that K21XXXX. where XXXX is your 4-digit Student Id.
3. Now compress that folder and submit on google-classroom.
4. Do not attach .exe file, otherwise it will show a threat or virus and not allow me to download.
5. Make sure you must Press the Turn-In button after uploading the solution folder. Otherwise, it will not be submitted.

#### **Task\_01:**

(Account Class) 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.

#### **Task - 02:**

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). 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.

#### **Task - 03:**

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 int). 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.

**Task - 04:**

Create a class called `Date` that includes three pieces of information as data members—a month (type `int`), a day (type `int`) and a year (type `int`). Your class should have constructor with three parameters that uses the parameters to initialize the three data members. For the purpose of this exercise, assume that the values provided for the year and day are correct, but ensure that the month value is in the range 1–12; if it isn't, set the month to 1. Provide a set and a get function for each data member. Provide a member function `displayDate` that displays the month, day and year separated by forward slashes (/). Write a test program that demonstrates class `Date`'s capabilities.

**Task - 05:**

(Target-Heart-Rate Calculator) While exercising, you can use a heart-rate monitor to see that your heart rate stays within a safe range suggested by your trainers and doctors. According to the American Heart Association (AHA) ([www.americanheart.org/presenter.jhtml?identifier=4736](http://www.americanheart.org/presenter.jhtml?identifier=4736)), the formula for calculating your maximum heart rate in beats per minute is 220 minus your age in years. Your target heart rate is a range that is 50–85% of your maximum heart rate. [Note: These formulas are estimates provided by the AHA. Maximum and target heart rates may vary based on the health, fitness and gender of the individual. Always consult a physician or qualified health care professional before beginning or modifying an exercise program.] Create a class called `HeartRates`. The class attributes should include the person's first name, last name and date of birth (consisting of separate attributes for the month, day and year of birth). Your class should have a constructor that receives this data as parameters. For each attribute provide set and get functions. The class also should include a function `getAge` that calculates and returns the person's age (in years), a function `getMaximumHeartRate` that calculates and returns the person's maximum heart rate and a function `getTargetHeartRate` that calculates and returns the person's target heart rate. Since you do not yet know how to obtain the current date from the computer, function `getAge` should prompt the user to enter the current month, day and year before calculating the person's age. Write an application that prompts for the person's information, instantiates an object of class `HeartRates` and prints the information from that object—including the person's first name, last name and date of birth—then calculates and prints the person's age in (years), maximum heart rate and target-heart-rate range.

### Task - 06:

A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed; otherwise "Required copies not in stock" is displayed.

Design a system using a class called books with suitable member functions and constructors.Task - 05:

Write a class called CoffeeShop, which has three instance variables:

Name: a string (basically, of the shop)

Menu: an array of items (of type MenuItem), with each item containing the item (name of the item), type (whether a food or a drink) and price.

Orders: an empty array and seven methods:

addOrder: adds the name of the item to the end of the orders array if it exists on the menu.

Otherwise, return "This item is currently unavailable!"

fulfillOrder: if the orders array is not empty, return "The {item} is ready!". If the orders array is empty, return "All orders have been fulfilled!"

listOrders: returns the list of orders taken, otherwise, an empty array.

dueAmount: returns the total amount due for the orders taken.

cheapestItem: returns the name of the cheapest item on the menu.

drinksOnly: returns only the item names of type drink from the menu.

foodOnly: returns only the item names of type food from the menu.

**Enjoy Coding**