EASTERN UNIVERSITY, SRI LANKA

DEPARTMENT OF MATHEMATICS

FIRST YEAR EXAMINATION IN SCIENCE - 2015/2016

SECOND SEMESTER (MAY/JUNE, 2018)

CS 153 - OBJECT ORIENTED PROGRAMMING TECHNIQUES

(PRACTICAL)

Answer all questions **Time allowed: Two Hours**

---

(1) Design a class named as *Rect* to represent a rectangle. The class contains:

- Two double data members named as *rectWidth* and *rectHeight* that specify the width and height of the rectangle.
- A default constructor that creates a default rectangle with width and height values as 1.
- A constructor that creates a rectangle with a specified width and height.
- A function named as *getArea()* that returns the area of a rectangle.
- A function named as *getPerimeter()* that returns the perimeter of a rectangle.
- A overloading function which overloads the (+) operator to add two *Rect* objects. Adding two *Rect* objects should give a new *Rect* object whose height is sum of the heights of the two *Rect* objects and width is sum of the widths of the two *Rect* objects.
- A overloading function which overloads the (<) operator to compare two *Rect* objects. The areas of the *Rect* objects are used to decide the bigger *Rect* object.

Write a test program that creates two *Rect* objects. Assign the values of width and height as 4 and 40 to the first object respectively and values of width and height as 4.5 and 35.9 to the second object respectively.

Call the functions mentioned above. Display the properties and output values of each object.

(2) Write a C++ program to design a class named as *Account* that contains:

- A private integer data member named as *id* for the *Account*.
- A private double data member named as *balance* for the *Account*.
- A private double data member named as *annualInterestRate* that stores the current interest rate. Assume all accounts have the same interest rate.
- A default constructor that creates an account with default values (*id*=0, *balance*=0 and *annualInterestRate*=0).
- A constructor that creates an account with the specified *id* and initial *balance*.
- The accessor and mutator methods for *id*, *balance*, and *annualInterestRate*.
- A method named as *getMonthlyInterestRate()* that returns the monthly interest rate.
- A method named as *getMonthlyInterest()* that returns the monthly interest.
- A method named as *withdraw* that withdraws a specified amount from the account.
- A method named as *deposit* that deposits a specified amount to the account.

(Hint: The method *getMonthlyInterest()* is to return monthly interest, not the interest rate. *monthlyInterestRate* is calculated by *annualInterestRate*/12. Monthly interest is calculated by *balance* * *monthlyInterestRate*).

Design a class *ATM* to simulate an ATM machine using the *Account* class, that contains:

- Three private integer data members *choice*, *withdrawAmount* and *depositAmount*.
- A method *menu()* that displays the menu of selection.
- A method *findChoice()* that reads the *id* as a parameter and prompts the user to enter a choice.

(Hint: Enter a choice 1 for viewing the current balance, 2 for withdrawing the money, 3 for depositing the money, and 4 for exiting the main menu.)

i. Write a test program that creates an *Account* object with an account id 1122, a *balance* of Rs. 20,000, and an *annualInterestRate* of 4.5%.

Use the *withdraw* method to withdraw Rs. 2,500, use the *deposit* method to deposit Rs. 3,000, and print the balance and the monthly interest.

ii. Create ten accounts in an array with *id* 0, 1, . . . , 9, and initial *balance* Rs. 500. Display the menu and display the values for the appropriate choices.

Sample Output for menu:

Enter an id: 5

Main menu

1: check balance

2: withdraw

3: deposit

4: exit

Enter a choice: 1 (if the choice is 1)

The balance is Rs. 500.0

Enter a choice: 2 (if the choice is 2)

Enter an amount to withdraw: Rs. 200

The balance is Rs. 300.0

Enter a choice: 3 (if the choice is 3)

Enter an amount to deposit: Rs. 200

The balance is Rs.700.0

Enter a choice: 4 (if the choice is 4)

Exit