Ahmedabad University
School of Engineering and Applied Science
B. Tech. (ICT) Semester – II
CSC103 – Object Oriented Programming Lab
Lab Assignment – 2

Assignment Date: January 28, 2019 **Last Date of Submission: February 8, 2019**

Topics covered: Classes, Objects, Constructors, Polymorphism, Array of Objects

Important Instructions:

Follow below coding and naming conventions while developing your programs:

1. Class names should be nouns and first letter must be capital letter. The first letter of each internal word capitalized. Use whole words-avoid acronyms and abbreviations E.g. BankAccount, Employee, AdminStaff, SavingAccount
2. Methods should be verbs with camel case, first letter lowercase and the first letter of each internal word capitalized.
   E.g. addEmployee( ); deleteEmployee( ); displayEmployeeList( );
3. Variables: Variable names should be meaningful. E.g. for Employee Class, variable names should be employeeId, employeeName, employeeDesignation, employeeScaleOfPay, etc.
4. Every Program should have header having following information in multi-line comments or java doc comments

```
/*
   @author RollNo. - Firstname Lastname                    @version dd/mm/yyyy
  Description: Write program definition.
*/
```

   Every Program should have footer with output of the Program in multi-line comment

```
/*  PROGRAM OUTPUT  */
```

Policy on Plagiarism (Copying of code/programs) and academic dishonesty

Copying Lab assignment/program code doesn't help you to learn the concepts.

Programs designed, developed and submitted by a student should be the result of original and individual work based on his/her own efforts. Full or part of the code should not be copied from internet or from peer students or other sources. A student should not share/circulate the code/programs developed by them (for individual assignments) with their peers in any form. Violation of above will be considered as academic dishonesty and any such case will be strictly dealt with and liable to get zero in the evaluation.

Exercises based on Classes, Objects, Constructors, Polymorphism, Array of Objects

This laboratory exercise helps you to understand the *basic concepts* of Object Oriented Programming.

1. Aim of this exercise is to understand how to define a class and create objects of the class.

Write a Java class **Student** to meet the following specification.

- The class should be able to support a 5 digit student ID, student name, marks for 3 subjects.
- Define methods to set and get each of the attributes, and calculate the average for the student.
- Write a StudentTest program to test Student class.
- Create 2 or 3 students object to check the Student class.

2. Write a Java program that defines the Circle class and uses it to create two circle objects. Design a **Circle** class based in following information:

- Two private instance variables: radius (with data type double) and color (with data type String), with default value of radius as 1.0 and color as "blue".
- Two *overloaded* constructors - a *default* constructor with no argument, and a constructor which takes a double argument for radius.
- Two public methods: getRadius() and getArea(), which return the radius and area of this instance, respectively.

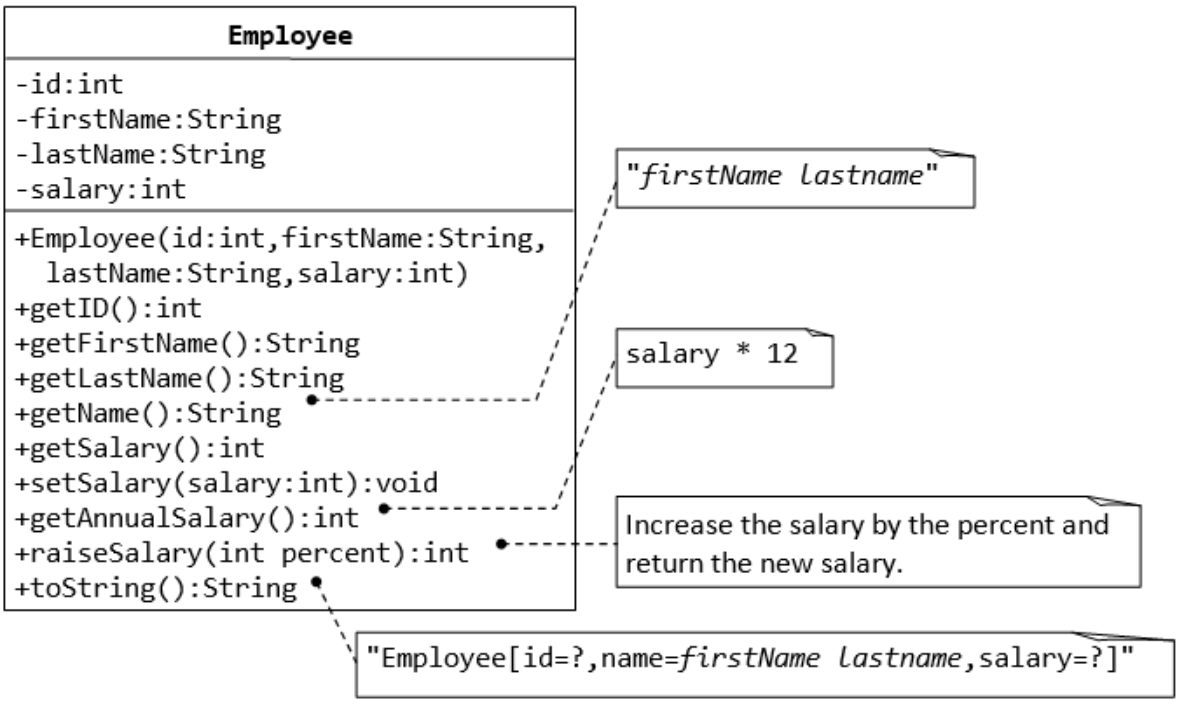
Create circle objects using different constructors (default and parameterized). Invoke public method getArea() for computing the area of circle.

3. Write a Java program that defines the Rectangle class and creates objects of it. Design a **Rectangle** class based in following information:

- Two private instance variables: length (with data type float) and width (with data type float), with default value of 1.0 to both variables.
- Two *overloaded* constructors - a *default* constructor with no argument, and a constructor which takes float arguments for length and width.
- Demonstrate use of getter and setter method for setting and displaying values of length and width.
- Define public methods: getArea() and getPerimeter() for computing the area and perimeter of rectangle respectively.

Create rectangle objects using different constructors (default and parameterized). Invoke public method getArea() for computing the area of rectangle.

4. Write a Java program which creates Employee class based on information given in the following Employee class diagram:



Instance Variables: id, firstname, lastname, salary
Constructor: Employee (int id, string firstname, string lastname, int salary)
Getter and Setter methods: getId(), getFirstName(), setSalary(), etc.
Public methods: int getAnualSalary(), int raiseSalary(int percent)

Write a *test program* called TestEmployee (in another source file called TestEmployee.java) which uses the Employee class. Demonstrate initialization of Employee object using constructor. Create array of Employee object. Invoke other methods of Employee to demonstrate its use.

5. Design a class named **Stock** that contains following information:

- A string data field named **symbol** for the stock's symbol.
- A string data field named **name** for the stock's name.
- A **double** data field named **previousClosingPrice** that stores the stock price for the previous day.
- A **double** data field named **currentPrice** that stores the stock price for the current time.
- A constructor that creates a stock with the specified symbol and name.

- A method named **getChangePercent()** that returns the percentage changed from **previousClosingPrice** to **currentPrice**.

Write a test program that creates a **Stock** object with the stock symbol **RIL**, the name **Reliance Industries**, and the previous closing price of **1050.5**. Set a new current price to **1150.55** and display the price-change percentage.

6. **Super Store:** Write a Java program to store items for any Super Store (like Reliance Mart) with details item no., name and price. Provide the functionality of inserting, deleting and updating the items. Also provide the functionality of searching item details based on item no. and name.

7. **Online Banking** – Write a Java program to define a BankAccount class to represent the savings bank account in a bank operated by customer. Define suitable data members and methods to operate online banking related transactions:

   Data members:
   - accountNo, customerName, balance

   Methods:
   - parameterized constructor
   - withdraw(int amount)
   - deposit(int amount)
   - depositInterest(float rate) - calculate simple interest based on balance and deposit it in the bank account.

   In main method, create a bank account object with suitable values and perform withdraw, deposit and interest calculation related transactions.

8. **Result Declaration System:** Write a program that asks the student his rollNumber(unique) and displays his subject-wise marks, total marks and percentage in B. Tech. Sem I examination. Consider following information while designing your program:

   - Create Student class and store marks related information in Student class.
   - Consider total number of students is 10. Solve the given problem using array of student objects.
   - Create different methods in Student class which calculates total marks, returns percentage.
   - Validations: If wrong rollNo is entered (not in array), display appropriate error message.