**Class oriented OOP Problem Statements**

1. Define the class TaxiMeter having the following members:

**Data Members/instance members**

int taxino – to store taxi number

String name - to store pessanger’s name

int km – to store number of kilometres travelled

**Member methods**

TaxiMeter() - Constructor to initialize taxino to 0, name to “” and km to 0

input() - to take input values of members

calculate() – to calculate the bill of a customer according to given conditions

|  |  |
| --- | --- |
| **Kilometers Travelled** | **Rate / km** |
| <= 1 | ₹ 25 |
| 1 < km <=6 | ₹ 10 |
| 6 < km <= 12 | ₹ 15 |
| 12 < km <= 18 | ₹ 20 |
| > 18 | ₹ 25 |

display() - to display the output in the following format

Taxi No. Name Km Travelled Bill Amount

Create an object in the main method and call the above methods to achieve the output.

1. Create a class Deposit and calculate the maturity amount with the following function
2. double termDeposit(int p, double r , int n) where p is principal, r is rate of interest and n is the time period in years. Calculate

**A = P (1 + r/100)**

1. double recurDeposit(int p, double r , int n) where p is monthly instalment , r is rate of interest and n is the time period in months. Calculate

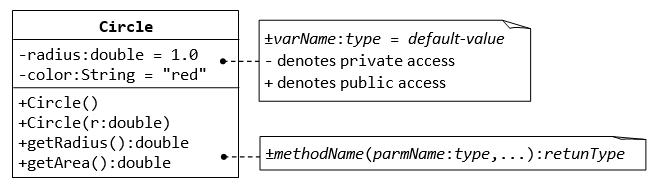
A = P x n + P x( n(n+1))/ 2 \* r/100 \*1/12

Write a main method to call appropriate methods

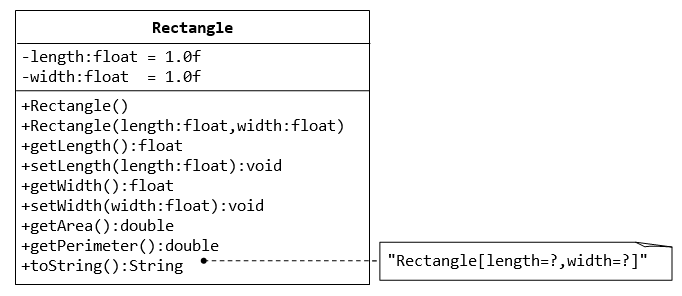
**Reference : ICSE board -> Bhaktivedanta Swami Mission School**

1. **A class called circle is designed as shown in the following class diagram. It contains:**

* Two private instance variables: radius (of the type double) and color (of the type String), with default value of 1.0 and "red", respectively.
* 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.

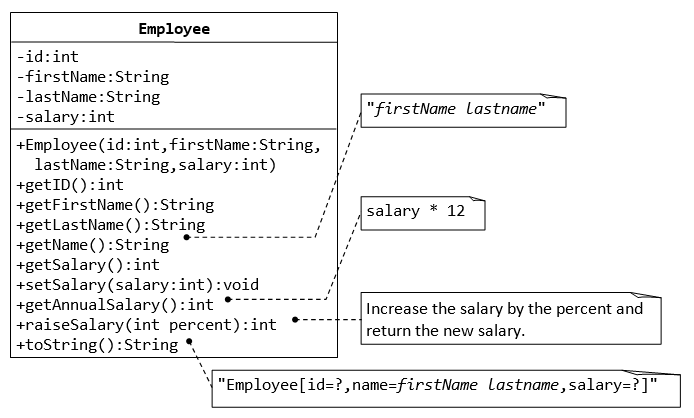


1. **The Rectangle Class**



1. **The Employee Class**

Create a class called Employee that includes four instance variables—an Id (Employee ID), a first name (type String), a last name (type String) and a monthly salary (double). Provide a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary is not positive, do not set its value. Write a test application named EmployeeTest that demonstrates class Employee’s capabilities. Create two Employee objects and display each object’s yearly salary. Then give each Employee a 10% raise and display each Employee’s yearly salary again.

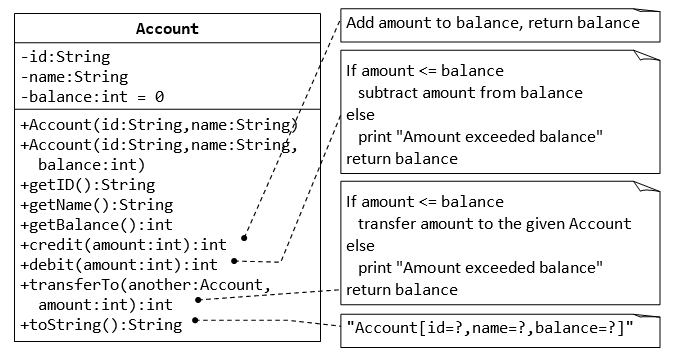


1. **Account Class**

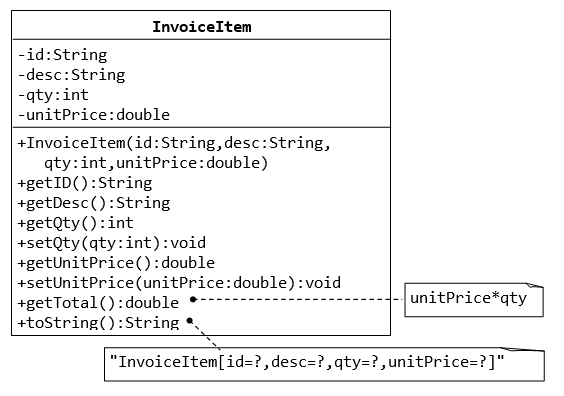
Create a class “Account” which contains account number, name and balance of a customer. Develop user defined methods credit() , debit( ) and getBalance( ). Ensure that the debit amount does not exceed the Account’s balance. If it does, the balance should be left unchanged and the method should print a message indicating "Debit amount exceeded account balance."

Test the program by entering customer details.

Extend the above program as per the requirements shown as below:

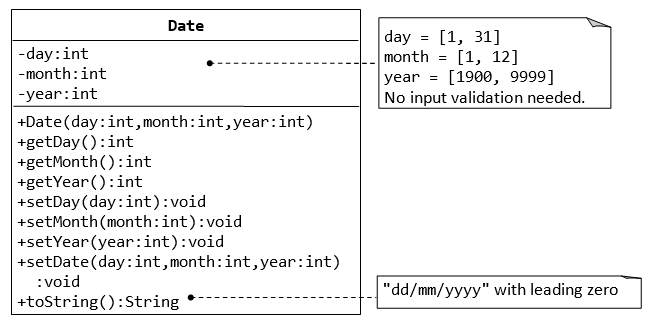


1. Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information as instance variables—a part number (type String), a part description (type String), a quantity of the item being purchased (type int) and a price per item (double). Your class should have a constructor that initializes the four instance variables. Provide a set and a get method for each instance variable. In addition, provide a method named getInvoiceAmount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as a double value.



1. Create a class called Date that includes three instance variables—a month (type int), a day (type int) and a year (type int). Provide a constructor that initializes the three instance variables and assumes that the values provided are correct. Provide a set and a get method for each instance variable. Provide a method displayDate that displays the month, day and year separated by forward slashes (/). Write a test application named DateTest that demonstrates class Date’s capabilities.

**Extension:** The user will enter the format of date from the keyboard and the application should display the date accordingly.



Note: Later we will put validation on the day, month and year input

Ref: https://www.ntu.edu.sg/home/ehchua/programming/java/J3f\_OOPExercises.html