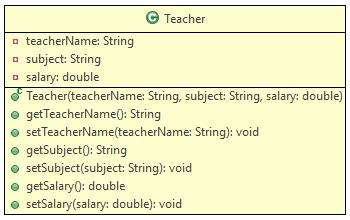
**Capstone project Day2 Assignment-1**

Implement the class Teacher based on the class diagram and description given below.



**Method Description**

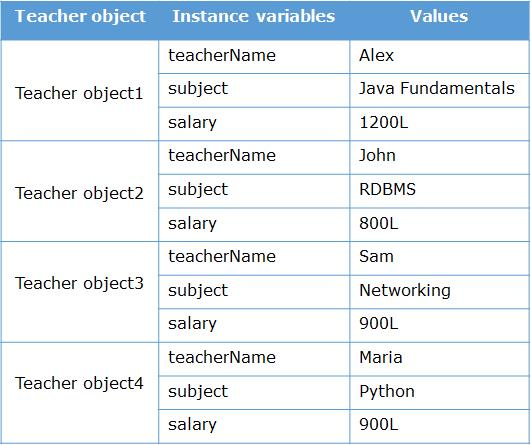
Teacher(String teacherName, String subject, double salary)

Initialize the values of all the instance variables appropriately with the values passed

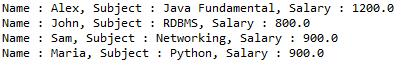
Create a Tester class. Create 4 objects of Teacher class. Create an array of type Teacher store the created objects and display the details of the teachers.

**Sample Input and Output**

**Input**



**Output**



**Solution:-**

package capstoneday2;

public class teacher {

private String teacherName;

private String subject;

private double salary;

public teacher(String teacherName, String subject, double salary) {

this.teacherName = teacherName;

this.subject = subject;

this.salary = salary;

}

public String getTeacherName() {

return teacherName;

}

public String getSubject() {

return subject;

}

public double getSalary() {

return salary;

}

}

//Tester class

package capstoneday2;

public class Tester {

public static void main(String[] args) {

// Create 4 objects of Teacher class

teacher teacher1 = new teacher("Alex", "java fundamental", 1200l);

teacher teacher2 = new teacher("Jahn", "Rdbms", 800l);

teacher teacher3 = new teacher("Sam", "Networking", 900l);

teacher teacher4 = new teacher("Maria", "python", 900l);

teacher[] teachers = new teacher[]{teacher1, teacher2, teacher3, teacher4};

for (int i = 0; i < teachers.length; i++) {

System.***out***.println("Teacher Name: " + teachers[i].getTeacherName());

System.***out***.println("Subject: " + teachers[i].getSubject());

System.***out***.println("Salary: " + teachers[i].getSalary());

System.***out***.println("-------------------------");

}

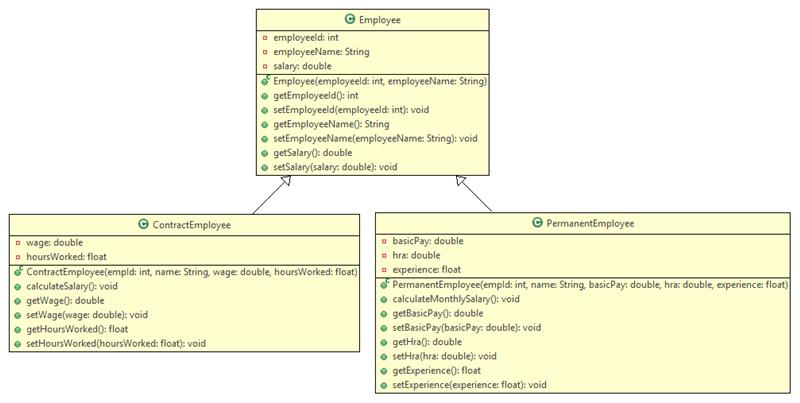
}

}

**Problem Statement**

A construction company wants to keep a record of the employees working in it. There are permanent employees as well as contract employees. Contract employees work on an hourly basis whereas permanent employees are paid monthly salary. An application needs to be developed for the company for storing the employee details.

Implement the classes based on the class diagram and description given below.



**Method Description**

**Employee**

**Employee(intemployeeId, String employeeName)**

* Initialize the employeeId and employeeName instance variables appropriately with the values passed to the constructor.

Implement the getter and setter methods appropriately.

**PermanentEmployee**

**PermanentEmployee(intempId, String name, double basicPay, double hra, float experience)**

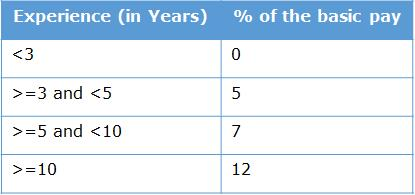
* Initialize the employeeId, employeeName, basicPay, hra and experience instance variables appropriately with the values passed to the constructor.

**calculateMonthlySalary()**

* Calculate the salary of the employee using the formula given below.

salary = basic pay + hra + variable component

* Variable component is calculated based on the employee's experience according to the table given below.



Implement the getter and setter methods appropriately.

**ContractEmployee**

**ContractEmployee(intempId, String name, double wage, float hoursWorked)**

* Initialize the employeeId, employeeName, wage and hoursWorked instance variables appropriately with the values passed to the constructor.

**calculateSalary()**

* Calculate the salary of the employee using the formula given below.

salary = hoursWorked \* wage

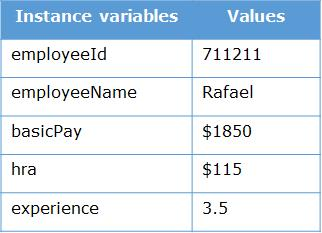
Implement the getter and setter methods appropriately.

Test the functionalities using the provided Tester class.

**Input and Output**

**For PermanentEmployee**

**Input**

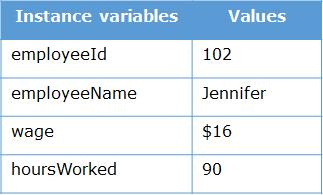


**Output**

IMG_262

**For ContractEmployee**

**Input**



**Output**

IMG_264

**Solution:-**

package employee.com;

public class Employee {

private int employeeId;

private String employeeName;

public Employee(int employeeId, String employeeName) {

this.employeeId = employeeId;

this.employeeName = employeeName;

}

public int getEmployeeId() {

return employeeId;

}

public void setEmployeeId(int employeeId) {

this.employeeId = employeeId;

}

public String getEmployeeName() {

return employeeName;

}

public void setEmployeeName(String employeeName) {

this.employeeName = employeeName;

}

}

//permanent employee class

package employee.com;

public class PermanentEmployee extends Employee {

private double basicPay;

private double hra;

private float experience;

public PermanentEmployee(int employeeId, String employeeName, double basicPay, double hra, float experience) {

super(employeeId, employeeName);

this.basicPay = basicPay;

this.hra = hra;

this.experience = experience;

}

public double calculateMonthlySalary() {

double variableComponent = 0.0;

if (experience < 5) {

variableComponent = 0.1 \* basicPay;

} else if (experience >= 5 && experience < 10) {

variableComponent = 0.2 \* basicPay;

} else {

variableComponent = 0.3 \* basicPay;

}

return basicPay + hra + variableComponent;

}

public double getBasicPay() {

return basicPay;

}

public void setBasicPay(double basicPay) {

this.basicPay = basicPay;

}

public double getHra() {

return hra;

}

public void setHra(double hra) {

this.hra = hra;

}

public float getExperience() {

return experience;

}

public void setExperience(float experience) {

this.experience = experience;

}

}

//contract employee class

package employee.com;

public class ContractEmployee extends Employee {

private double wage;

private float hoursWorked;

public ContractEmployee(int employeeId, String employeeName, double wage, float hoursWorked) {

super(employeeId, employeeName);

this.wage = wage;

this.hoursWorked = hoursWorked;

}

public double calculateSalary() {

return hoursWorked \* wage;

}

public double getWage() {

return wage;

}

public void setWage(double wage) {

this.wage = wage;

}

public float getHoursWorked() {

return hoursWorked;

}

public void setHoursWorked(float hoursWorked) {

this.hoursWorked = hoursWorked;

}

}

//Tester class

package employee.com;

public class Tester {

public static void main(String[] args) {

PermanentEmployee permanentEmployee = new PermanentEmployee(711211, "Rafael", 50000.0, 10000.0, 3.5f);

System.***out***.println("Hi Rafeal,your Salary is: " + permanentEmployee.calculateMonthlySalary());

ContractEmployee contractEmployee = new ContractEmployee(102, "Jennifer", 300.0, 90.0f);

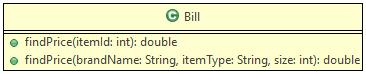
System.***out***.println("Hi Jennifer,your Salary is: " + contractEmployee.calculateSalary());

}

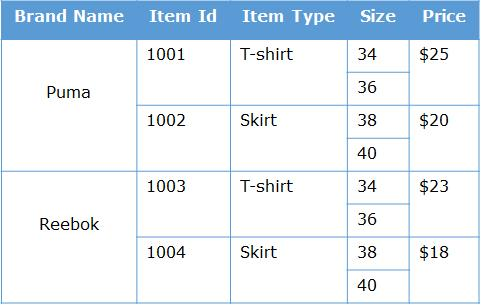
}

**Problem Statement**

The Bill class is used to find the price of items for calculation. Implement a class Bill based on the class diagram and description given below.



The details of the items are given below.



**Method Description**

**findPrice(intitemId)**

* Find and return the price based on the itemId using the table given above.
* If the itemId passed to method is invalid, return the price as 0.

**findPrice(String brandName, String itemType, int size)**

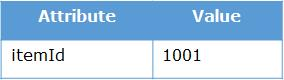
* Find and return the price based on the brandName, itemType and size using the table given above.
* If any invalid details are passed to the method, return the price as 0.

Test the functionalities using the provided Tester class.

**Sample Input and Output**

For findPrice(intitemId)

**Input**

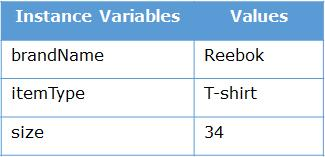


**Output**

IMG_268

For findPrice(String brandName, String itemType, int size)

**Input**



**Output**

IMG_270

**Solution:-**

package bill.com;

class Bill {

public double findPrice(int itemId) {

switch (itemId) {

case 1001: return 25.0;

case 1002: return 20.0;

case 1003: return 23.0;

case 1004: return 18.0;

default: return 0.0;

}

}

public double findPrice(String brandName, String itemType, int size) {

if ("puma".equalsIgnoreCase(brandName) && "T-shirt".equalsIgnoreCase(itemType) && size == 34) {

return 25.0;

} else if ("puma".equalsIgnoreCase(brandName) && "Skirt".equalsIgnoreCase(itemType) && size == 38) {

return 20.0;

} else if ("puma".equalsIgnoreCase(brandName) && "Skirt".equalsIgnoreCase(itemType) && size == 40) {

return 20.0;

} else if ("Reebok".equalsIgnoreCase(brandName) && "T-shirt".equalsIgnoreCase(itemType) && size == 34) {

return 23.0;

} else if ("Reebok".equalsIgnoreCase(brandName) && "Skirt".equalsIgnoreCase(itemType) && size == 36) {

return 18.0;

} else if ("Reebok".equalsIgnoreCase(brandName) && "Skirt".equalsIgnoreCase(itemType) && size == 38) {

return 18.0;

} else if ("Reebok".equalsIgnoreCase(brandName) && "Skirt".equalsIgnoreCase(itemType) && size == 40) {

return 18.0;

} else {

return 0.0;

}

}

}

//tester class

package bill.com;

import java.util.Scanner;

public class Tester {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.***in***);

Bill bill = new Bill();

// Test findPrice(int itemId)

System.***out***.println("Enter itemId:");

int itemId = scanner.nextInt();

double priceById = bill.findPrice(itemId);

System.***out***.println("Price of the selected item is $" + priceById);

// Test findPrice(String brandName, String itemType, int size)

scanner.nextLine(); // Consume newline

System.***out***.println("Enter brandName:");

String brandName = scanner.nextLine();

System.***out***.println("Enter itemType:");

String itemType = scanner.nextLine();

System.***out***.println("Enter size:");

int size = scanner.nextInt();

double priceByDetails = bill.findPrice(brandName, itemType, size);

System.***out***.println("Price of the selected item is $" + priceByDetails);

scanner.close();

}

}