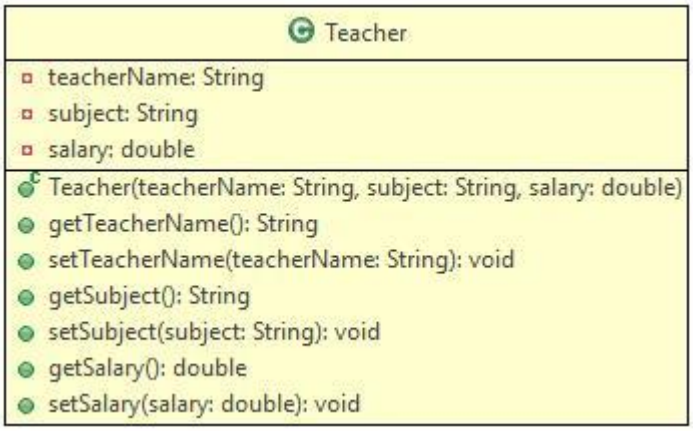
Capstone project Day2 Assignment-1

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

Teacher object	Instance variables	Values
Teacher object1	teacherName	Alex
	subject	Java Fundamentals
	salary	1200L
Teacher object2	teacherName	John
	subject	RDBMS
	salary	800L
Teacher object3	teacherName	Sam
	subject	Networking
	salary	900L
Teacher object4	teacherName	Maria
	subject	Python
	salary	900L

Output

```
Name : Alex, Subject : Java Fundamental, Salary : 1200.0
Name : John, Subject : RDBMS, Salary : 800.0
Name : Sam, Subject : Networking, Salary : 900.0
Name : Maria, Subject : Python, Salary : 900.0
```

Answer:

```
import java.util.Scanner;

class Assignment1 {

 private String teacherName;

 private String subject;

 private double salary;

 public Assignment1(String teacherName, String subject, double salary) {

    this.teacherName = teacherName;

    this.subject = subject;

    this.salary = salary;

 }

 public String getTeacherName() {

    return teacherName;

 }

 public void setTeacherName(String teacherName) {

    this.teacherName = teacherName;

 }

 public String getSubject() {

    return subject;

 }

 public void setSubject(String subject) {

    this.subject = subject;
```

```
    }
    public double getSalary() {
        return salary;
    }
    public void setSalary(double salary) {
        this.salary = salary;
    }
}
public class Assignment1Imp {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of teachers: ");
        int numTeachers = scanner.nextInt();
        scanner.nextLine();
        Assignment1[] teachers = new Assignment1[numTeachers];
        for (int i = 0; i < teachers.length; i++) {
            System.out.println("\nEnter details for Teacher " + (i + 1) + ":");
            System.out.print("Name: ");
            String name = scanner.nextLine();
            System.out.print("Subject: ");
            String subject = scanner.nextLine();
            System.out.print("Salary: ");
            double salary = scanner.nextDouble();
            scanner.nextLine();
            teachers[i] = new Assignment1(name, subject, salary);
```

```
        }
        System.out.println("\nDetails of Teachers:");
        for (Assignment1 teacher : teachers) {
            System.out.println("Name: " + teacher.getTeacherName() +
                    ", Subject: " + teacher.getSubject() +
                    ", Salary: " + teacher.getSalary());
        }
        scanner.close();
    }
}
```

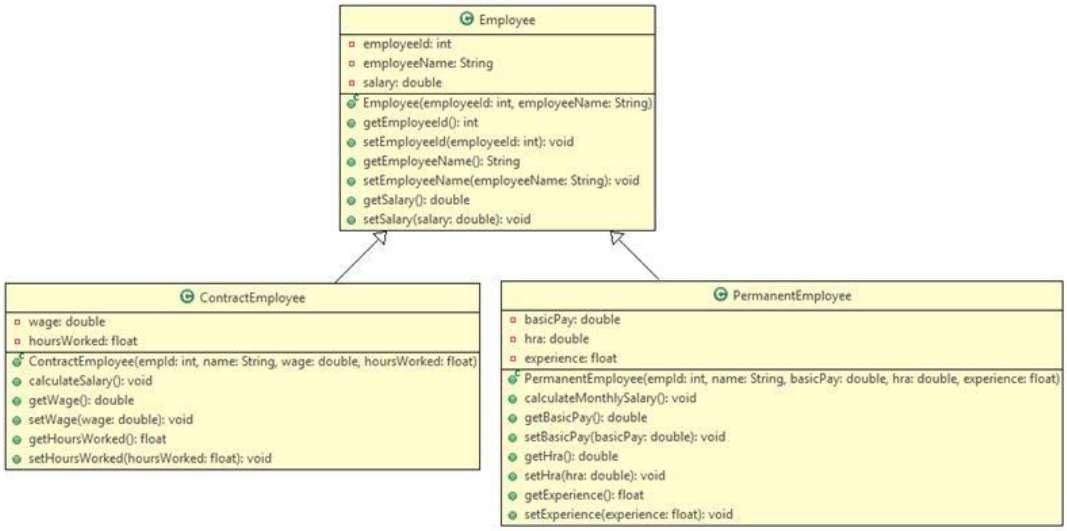
```
C:\Users\Administrator\Documents\Capstone_Project_Day2>javac Assignment1Imp.java

C:\Users\Administrator\Documents\Capstone_Project_Day2>java Assignment1Imp
Enter the number of teachers: 4

Enter details for Teacher 1:
Name: Alex
Subject: Java Fundamentals
Salary: 1200

Enter details for Teacher 2:
Name: John
Subject: RDBMS
Salary: 800

Enter details for Teacher 3:
Name: Sam
Subject: Networking
Salary: 900

Enter details for Teacher 4:
Name: Maria
Subject: Python
Salary: 900

Details of Teachers:
Name: Alex, Subject: Java Fundamentals, Salary: 1200.0
Name: John, Subject: RDBMS, Salary: 800.0
Name: Sam, Subject: Networking, Salary: 900.0
Name: Maria, Subject: Python, Salary: 900.0
```

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.

Experience (in Years)	% of the basic pay
<3	0
>=3 and <5	5
>=5 and <10	7
>=10	12

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

Instance variables	Values
employeeId	711211
employeeName	Rafael
basicPay	$1850
hra	$115
experience	3.5

Output

```
Hi Rafael, your salary is $2057.5
```

For ContractEmployee

Input

Instance variables	Values
employeeId	102
employeeName	Jennifer
wage	$16
hoursWorked	90

Output

```
Hi Jennifer, your salary is $1440.0
```

Answer:

```
import java.util.Scanner;


class Employee {
```

```
private int employeeId;
private String employeeName;
private double salary;

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;
}
```

```
    public double getSalary() {
        return salary;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }
}

class PermanentEmployee extends Employee {
    private double basicPay;
    private double hra;
    private float experience;

    public PermanentEmployee(int empId, String name, double basicPay, double hra,
float experience) {
        super(empId, name);
        this.basicPay = basicPay;
        this.hra = hra;
        this.experience = experience;
    }

    public void calculateMonthlySalary() {
        double variableComponent = 0;
        if (getExperience() < 3) {
            variableComponent = 0;
```

```
    } else if (getExperience() >= 3 && getExperience() < 5) {
        variableComponent = 0.05 * getBasicPay();
    } else if (getExperience() >= 5 && getExperience() < 10) {
        variableComponent = 0.07 * getBasicPay();
    } else if (getExperience() >= 10) {
        variableComponent = 0.12 * getBasicPay();
    }
    setSalary(getBasicPay() + getHra() + 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;
    }
}

class ContractEmployee extends Employee {
    private double wage;
    private float hoursWorked;

    public ContractEmployee(int empId, String name, double wage, float hoursWorked)
{
        super(empId, name);
        this.wage = wage;
        this.hoursWorked = hoursWorked;
    }

    public void calculateSalary() {
        setSalary(getHoursWorked() * getWage());
    }

    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;
    }
}

public class Assignment2 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Input for PermanentEmployee
        System.out.println("For PermanentEmployee:\n");
        System.out.print("Employee ID: ");
        int permEmpId = scanner.nextInt();
        scanner.nextLine();  // Consume newline
```

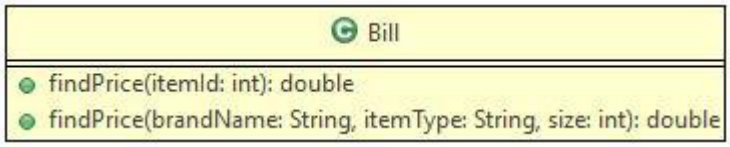
```
        System.out.print("Employee Name: ");
        String permEmpName = scanner.nextLine();
        System.out.print("Basic Pay: ");
        double basicPay = scanner.nextDouble();
        System.out.print("HRA: ");
        double hra = scanner.nextDouble();
        System.out.print("Experience: ");
        float experience = scanner.nextFloat();

        PermanentEmployee permanentEmployee = new
PermanentEmployee(permEmpId, permEmpName, basicPay, hra, experience);
        permanentEmployee.calculateMonthlySalary();
        System.out.println("Hi " + permanentEmployee.getEmployeeName() + ", your
salary is $" + permanentEmployee.getSalary());

        // Input for ContractEmployee
        System.out.println("For ContractEmployee: \n");
        System.out.print("Employee ID: ");
        int contractEmpId = scanner.nextInt();
        scanner.nextLine();  // Consume newline
        System.out.print("Employee Name: ");
        String contractEmpName = scanner.nextLine();
        System.out.print("Wage: ");
        double wage = scanner.nextDouble();
        System.out.print("Hours Worked: ");
        float hoursWorked = scanner.nextFloat();
```

```
        ContractEmployee contractEmployee = new ContractEmployee(contractEmpId,
contractEmpName, wage, hoursWorked);

        contractEmployee.calculateSalary();

        System.out.println("Hi " + contractEmployee.getEmployeeName() + ", your
salary is $" + contractEmployee.getSalary());



        scanner.close();

    }

}
```

```
C:\Users\Administrator\Documents\Capstone_Project_Day2>javac Assignment2.java

C:\Users\Administrator\Documents\Capstone_Project_Day2>java Assignment2
For PermanentEmployee:

Employee ID: 711211
Employee Name: Rafael
Basic Pay: 850
HRA: 115
Experience: 3.5
Hi Rafael, your salary is $1007.5
For ContractEmployee:

Employee ID: 102
Employee Name: Jennifer
Wage: 16
Hours Worked: 90
Hi Jennifer, your salary is $1440.0
```

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.

Brand Name	Item Id	Item Type	Size	Price
Puma	1001	T-shirt	34	$25
			36	
	1002	Skirt	38	$20
			40	
Reebok	1003	T-shirt	34	$23
			36	
	1004	Skirt	38	$18
			40	

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

Attribute	Value
itemId	1001

Output

```
Price of the selected item is $25.0
```

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

Input

Instance Variables	Values
brandName	Reebok
itemType	T-shirt
size	34

Output

```
Price of the selected item is $23.0
```

Answer:

```
import java.util.Scanner;

class Assignment3 {

   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;

        }

    }

}

public class Assignment3Imp {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        Assignment3 assignment3 = new Assignment3();
```

```
        System.out.println("Enter itemId:");
        int itemId = scanner.nextInt();
        double priceById = assignment3.findPrice(itemId);
        System.out.println("Price of the selected item is $" + priceById);
        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 = assignment3.findPrice(brandName, itemType, size);
        System.out.println("Price of the selected item is $" + priceByDetails);
    }
}
```