**ASSIGNMENT 2**

- **PROBLEM STATEMENT:**

1. **Write a Java program to create an Employee class to manage employee's information such as employee id, name, date of birth, designation and basic salary. Use parameterized constructor to initialize object of Employee class. Define printData0 methods to display employee details and Net Salary. The Net Salary is sum of basic salary + HRA + DA, where HRA is 20% of basic salary and DA is 90% of basic salary.**

- **SOURCE CODE:**

```
class Employee {
   int empId;
   String name,dob,designation;
   double basicSalary;

   Employee(int empId, String name, String dob, String designation, double basicSalary) {
      this.empId = empId;
      this.name = name;
      this.dob = dob;
      this.designation = designation;
      this.basicSalary = basicSalary;
   }

   double calculateNetSalary() {
      double hra = 0.20 * basicSalary;
      double da = 0.90 * basicSalary;
      return basicSalary + hra + da;
   }

   void printData() {
      System.out.println("Employee ID: " + empId);
      System.out.println("Name: " + name);
      System.out.println("Date of Birth: " + dob);
      System.out.println("Designation: " + designation);
      System.out.println("Basic Salary: " + basicSalary);
      System.out.println("Net Salary: " + calculateNetSalary());
   }
}

class p1{
   public static void main(String[] args) {
      Employee emp = new Employee (1,"SAYAK","02/08/2004","Software Engineer", 50000);
      emp.printData();
   }
}
```

- **OUTPUT:**

```
Employee ID: 1
Name: SAYAK
Date of Birth: 02/08/2004
Designation: Software Engineer
Basic Salary: 50000.0
Net Salary: 105000.0
```

- **PROBLEM STATEMENT:**

2. **Create a class Shape with methods calc_perimeter() and calc_area(). Derive two classes Circle(data member: radius) and Rectangle(data member: length & breadth) from it. Use constructor for objectinitialization. Calculate perimeter and area for both the shapes. (Use of method overriding).**

- **SOURCE CODE:**

```
abstract class Shape {
   abstract double calcPerimeter();
   abstract double calcArea();
}

class Circle extends Shape {
   double radius;

   Circle(double radius) { this.radius = radius; }

   public double calcPerimeter() { return 2 * Math.PI * radius; }

   public double calcArea() { return Math.PI * radius * radius; }
}

class Rectangle extends Shape {
   double length, breadth;

   Rectangle(double length, double breadth) {
      this.length = length;
      this.breadth = breadth;
   }

   public double calcPerimeter() { return 2 * (length + breadth); }

   public double calcArea() { return length * breadth; }
}

public class p2 {
   public static void main(String[] args) {
      Shape circle = new Circle(7);
      Shape rectangle = new Rectangle(5, 5);

      System.out.println("Circle Perimeter: " + circle.calcPerimeter());
      System.out.println("Circle Area: " + circle.calcArea());
      System.out.println("Rectangle Perimeter: " + rectangle.calcPerimeter());
      System.out.println("Rectangle Area: " + rectangle.calcArea());
   }
}
```

- **OUTPUT:**

```
Circle Perimeter: 43.982297150257104
Circle Area: 153.93804002589985
Rectangle Perimeter: 20.0
Rectangle Area: 25.0
```

- **PROBLEM STATEMENT:**

3. **Write a Java program to create a super class Vehicle having data member: regNo, companyName and price. Derive two different classes LightMotorVehicle (members-mileage) and HeavyMotorVehicle (members-capacity-in-tons). Accept the information for both light and heavy motor vehicles using parameterized getData() method and display the information using displayDetails( ) method. (use of super keyword in method overriding).**

- **SOURCE CODE:**

```
class Vehicle {
   String regNo;
   String companyName;
   double price;

   void getData(String regNo, String companyName, double price) {
      this.regNo = regNo;
      this.companyName = companyName;
      this.price = price;
   }

   void displayDetails() {
      System.out.println("Registration No: " + regNo);
      System.out.println("Company Name: " + companyName);
      System.out.println("Price: $" + price);
   }
}

class LightMotorVehicle extends Vehicle {
   double mileage;

   void getData(String regNo, String companyName, double price, double mileage) {
      super.getData(regNo, companyName, price);
      this.mileage = mileage;
   }

   void displayDetails() {
      super.displayDetails();
      System.out.println("Mileage: " + mileage + " km/l");
   }
}

class HeavyMotorVehicle extends Vehicle {
   double capacity;

   void getData(String regNo, String companyName, double price, double capacity) {
      super.getData(regNo, companyName, price);
      this.capacity = capacity;
   }

   void displayDetails() {
      super.displayDetails();
      System.out.println("Capacity: " + capacity + " tons");
   }
}
```

```
public class p3 {
   public static void main(String[] args) {

       LightMotorVehicle lightVehicle = new LightMotorVehicle();
       lightVehicle.getData("936NFJ", "TATA", 50000, 18.7);
       System.out.println("Light Motor Vehicle Details:");
       lightVehicle.displayDetails();
       System.out.println();

       HeavyMotorVehicle heavyVehicle = new HeavyMotorVehicle();
       heavyVehicle.getData("894AHE", "Volvo", 90000, 15.0);
       System.out.println("Heavy Motor Vehicle Details:");
       heavyVehicle.displayDetails();
}
```

- **OUTPUT:**

```
Light Motor Vehicle Details:
Registration No: 936NFJ
Company Name: TATA
Price: $50000.0
Mileage: 18.7 km/l

Heavy Motor Vehicle Details:
Registration No: 894AHE
Company Name: Volvo
Price: $90000.0
Capacity: 15.0 tons
```

- **PROBLEM STATEMENT:**

**4. Define a class Staff with members name and address. Define two sub-classes of Staff lass FullTimeStaff (department, salary) and PartTimeStaff (number_of_hours, rate_perhour). Define appropriate parameterized constructors for parent as well as child lass. Create one object for each FullTimeStaff and PartTimeStaff class and then Display details of the "FullTimeStaff and PartTimeStaff` objects.**

- **SOURCE CODE:**

```
class Staff {
   String name;
   String address;

   Staff(String name, String address) {
      this.name = name;
      this.address = address;
   }

   void displayDetails() {
      System.out.println("Name: " + name);
      System.out.println("Address: " + address);
   }
}

class FullTimeStaff extends Staff {
   String department;
   double salary;

   FullTimeStaff(String name, String address, String department, double salary) {
      super(name, address);
      this.department = department;
      this.salary = salary;
   }

   void displayDetails() {
      super.displayDetails();
      System.out.println("Department: " + department);
      System.out.println("Salary: $" + salary);
   }
}

class PartTimeStaff extends Staff {
   int numberOfHours;
   double ratePerHour;

   PartTimeStaff(String name, String address, int numberOfHours, double ratePerHour) {
      super(name, address);
      this.numberOfHours = numberOfHours;
      this.ratePerHour = ratePerHour;
   }
```

```
 void displayDetails() {
        super.displayDetails();
        System.out.println("Number of Hours: " + numberOfHours);
        System.out.println("Rate Per Hour: $" + ratePerHour);
    }
}

class p4 {
    public static void main(String[] args) {
        FullTimeStaff fullTimeStaff = new FullTimeStaff("Sohan", "Mecheda", "CSE", 60000);
        System.out.println("Full Time Staff Details:");
        fullTimeStaff.displayDetails();
        System.out.println();

        PartTimeStaff partTimeStaff = new PartTimeStaff("Ganguli", "Ramtarak", 20, 15);
        System.out.println("Part Time Staff Details:");
        partTimeStaff.displayDetails();
}
```

- **OUTPUT:**

```
Full Time Staff Details:
Name: Sohan
Address: Mecheda
Department: CSE
Salary: $60000.0

Part Time Staff Details:
Name: Ganguli
Address: Ramtarak
Number of Hours: 20
Rate Per Hour: $15.0
```

- **PROBLEM STATEMENT:**

5. **Write a java code to accept personal information like name, address, age for Person class and create Student & Faculty two subclasses of Person. Student class has two more states roll & department and Faculty class also has another two states major_subject & publication. Student class is further divided into ExStudent with attributes (organizationName, officeEmail) and CurrentStudentwith attributes (semester, sgpa). Create an object for each class ExStudent, CurrentStudent and Faculty and display all details about the ExStudent, CurrentStudent and Faculty.**

- **SOURCE CODE:**

```
class Person {
   String name;
   String address;
   int age;

   Person(String name, String address, int age) {
      this.name = name;
      this.address = address;
      this.age = age;
   }

   void displayDetails() {
      System.out.println("Name: " + name);
      System.out.println("Address: " + address);
      System.out.println("Age: " + age);
   }
}

class Student extends Person {
   String roll;
   String department;

   Student(String name, String address, int age, String roll, String department) {
      super(name, address, age);
      this.roll = roll;
      this.department = department;
   }

   void displayDetails() {
      super.displayDetails();
      System.out.println("Roll: " + roll);
      System.out.println("Department: " + department);
   }
}

class Faculty extends Person {
   String majorSubject;
   String publication;

   Faculty(String name, String address, int age, String majorSubject, String publication) {
      super(name, address, age);
      this.majorSubject = majorSubject;
      this.publication = publication;
```

```
    }

    void displayDetails() {
        super.displayDetails();
        System.out.println("Major Subject: " + majorSubject);
        System.out.println("Publication: " + publication);
    }
}

class ExStudent extends Student {
    String organizationName;
    String officeEmail;

    ExStudent(String name, String address, int age, String roll, String department, String
organizationName, String officeEmail) {
        super(name, address, age, roll, department);
        this.organizationName = organizationName;
        this.officeEmail = officeEmail;
    }

    public void displayDetails() {
        super.displayDetails();
        System.out.println("Organization Name: " + organizationName);
        System.out.println("Office Email: " + officeEmail);
    }
}

class CurrentStudent extends Student {
    int semester;
    double sgpa;

    CurrentStudent(String name, String address, int age, String roll, String department, int semester,
double sgpa) {
        super(name, address, age, roll, department);
        this.semester = semester;
        this.sgpa = sgpa;
    }

    void displayDetails() {
        super.displayDetails();
        System.out.println("Semester: " + semester);
        System.out.println("SGPA: " + sgpa);
    }
}

class p5 {
    public static void main(String[] args) {
        ExStudent exStudent = new ExStudent("Manas", "Kolaghat", 18, "29", "Computer Science",
"GateIt", "partha@GateIt.com");
        System.out.println("Ex Student Details:");
        exStudent.displayDetails();
        System.out.println();

        CurrentStudent currentStudent = new CurrentStudent("Anwesha", "Mecheda", 18, "30",
"CSE", 5, 9.9);
```

```
        System.out.println("Current Student Details:");
        currentStudent.displayDetails();
        System.out.println();

        Faculty faculty = new Faculty("Partha", "Ramtarak", 18, "Physics", "Quantum Mechanics");
        System.out.println("Faculty Details:");
        faculty.displayDetails();
    }
}
```

- **OUTPUT:**

```
Ex Student Details:
Name: Manas
Address: Kolaghat
Age: 18
Roll: 29
Department: Computer Science
Organization Name: GateIt
Office Email: partha@GateIt.com

Current Student Details:
Name: Anwesha
Address: Mecheda
Age: 18
Roll: 30
Department: CSE
Semester: 5
SGPA: 9.9

Faculty Details:
Name: Partha
Address: Ramtarak
Age: 18
Major Subject: Physics
Publication: Quantum Mechanics
```