PERSON.JAVA

```
package person.inherit;

import javax.persistence.DiscriminatorColumn;
import javax.persistence.DiscriminatorType;
import javax.persistence.DiscriminatorValue;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Inheritance;
import javax.persistence.InheritanceType;

@Entity
@Inheritance(strategy = InheritanceType.TABLE_PER_CLASS)
@DiscriminatorColumn(name = "PERSON", discriminatorType = DiscriminatorType.STRING)
@DiscriminatorValue("person")

public class Person {
        @Id
        @GeneratedValue
        private int id;
        private String name;
        private String address;
        private String department;

        public Person() {
                super();
                // TODO Auto-generated constructor stub
        }

        public Person(String name, String address, String department) {
                super();
                this.name = name;
                this.address = address;
                this.department = department;
        }

        public int getId() {
                return id;
        }

        public void setId(int id) {
                this.id = id;
        }
```

```
	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}

	public String getAddress() {
		return address;
	}

	public void setAddress(String address) {
		this.address = address;
	}

	public String getDepartment() {
		return department;
	}

	public void setDepartment(String department) {
		this.department = department;
	}

}
```

TEACHER.JAVA

```
package person.inherit;

import javax.persistence.Column;
import javax.persistence.DiscriminatorValue;
import javax.persistence.Entity;
import javax.persistence.Transient;

@Entity
@DiscriminatorValue("teacher")
public class Teacher extends Person {

	private double basicSalary;
	@Column(name = "HRA", nullable = false)
	private float houseRentAllowance;
	@Column(name = "TA", nullable = false)
	private float travelAllowance;
	@Column(name = "DA", nullable = false)
	private float dearnessAllowance;
	@Column(name = "PF", nullable = false)
```

```
    }

    private float providentFund;
    @Transient
    private double netSalary;

    public Teacher() {
        super();
    }

    public Teacher(String name, String address, String department, double basicSalary, float
houseRentAllowance,
            float travelAllowance, float dearnessAllowance, float providentFund) {
        super(name, address, department);
        this.basicSalary = basicSalary;
        this.houseRentAllowance = houseRentAllowance;
        this.travelAllowance = travelAllowance;
        this.dearnessAllowance = dearnessAllowance;
        this.providentFund = providentFund;
    }

    public double getBasicSalary() {
        return basicSalary;
    }

    public void setBasicSalary(double basicSalary) {
        this.basicSalary = basicSalary;
    }

    public float getHouseRentAllowance() {
        return houseRentAllowance;
    }

    public void setHouseRentAllowance(float houseRentAllowance) {
        this.houseRentAllowance = houseRentAllowance;
    }

    public float getTravelAllowance() {
        return travelAllowance;
    }

    public void setTravelAllowance(float travelAllowance) {
        this.travelAllowance = travelAllowance;
    }

    public float getDearnessAllowance() {
        return dearnessAllowance;
    }

    public void setDearnessAllowance(float dearnessAllowance) {
        this.dearnessAllowance = dearnessAllowance;
    }

    public float getProvidentFund() {
        return providentFund;
    }

    public void setProvidentFund(float providentFund) {
        this.providentFund = providentFund;
    }

    public double calculate(double basicSalary, float houseRentAllowance, float travelAllowance,
            float dearnessAllowance, float providentFund) {
        netSalary = this.basicSalary
                + (((this.houseRentAllowance + this.travelAllowance +
this.dearnessAllowance - this.providentFund)
                        / 100) * this.basicSalary);
        return netSalary;
    }

}
```

STUDENT.JAVA

```
package person.inherit;

import javax.persistence.DiscriminatorValue;
import javax.persistence.Entity;

@Entity
@DiscriminatorValue("student")
public class Student extends Person {
    private int creditsEarned;
    private String totalGrade;

    public Student() {
        super();
        // TODO Auto-generated constructor stub
    }

    public Student(String name, String address, String department, int creditsEarned, String
totalGrade) {
        super(name, address, department);
        this.creditsEarned = creditsEarned;
```

```
                this.totalGrade = totalGrade;
        }

        public int getCreditsEarned() {
                return creditsEarned;
        }

        public void setCreditsEarned(int creditsEarned) {
                this.creditsEarned = creditsEarned;
        }

        public String getTotalGrade() {
                return totalGrade;
        }

        public void setTotalGrade(String totalGrade) {
                this.totalGrade = totalGrade;
        }

}
```

SOLUTIONPERSON.JAVA

```
package org.main;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

import person.inherit.Person;
import person.inherit.Student;
import person.inherit.Teacher;

public class SolutionPerson {

        public static void main(String[] args) throws IOException {
                SessionFactory sf = new Configuration().configure().buildSessionFactory();
                Session session = sf.openSession();
                BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));
                session.beginTransaction();

                String name, address, department;
                System.out.println("Do you want to enter teacher or student
details:\n1.Teacher\n2.Student");
                int ch = Integer.valueOf(bf.readLine());

                switch (ch) {
                case 1:
                        System.out.println("Enter the name:");
                        name = bf.readLine();
                        System.out.println("Enter the address:");
                        address = bf.readLine();
                        System.out.println("Enter the department:");
                        department = bf.readLine();
                        System.out.print("enter Basic salary:");
                        double basicSalary = Double.valueOf(bf.readLine());

                        System.out.print("enter HRA:");
                        float houseRentAllowance = Float.valueOf(bf.readLine());

                        System.out.print("enter TA:");
                        float travelAllowance = Float.valueOf(bf.readLine());

                        System.out.print("enter DA:");
                        float dearnessAllowance = Float.valueOf(bf.readLine());

                        System.out.print("enter PF:");
                        Float providentFund = Float.valueOf(bf.readLine());
                        Teacher teach = new Teacher();
                        teach.setName(name);
                        teach.setAddress(address);
                        teach.setDepartment(department);
                        teach.setBasicSalary(basicSalary);
                        teach.setHouseRentAllowance(houseRentAllowance);
                        teach.setTravelAllowance(travelAllowance);
                        teach.setDearnessAllowance(dearnessAllowance);
                        teach.setProvidentFund(providentFund);
                        System.out.println("total net salary=" + teach.calculate(basicSalary,
houseRentAllowance, travelAllowance,
                                        dearnessAllowance, providentFund));
                        System.out.println("Staff details saved");
                        session.save(teach);
                        break;
                case 2:
                        System.out.println("Enter the name:");
                        name = bf.readLine();
                        System.out.println("Enter the address:");
```

```
            address = bf.readLine();
            System.out.println("Enter the department:");
            department = bf.readLine();
            System.out.println("Enter the credits earned:");
            int credits = Integer.valueOf(bf.readLine());
            System.out.println("total grade:");
            String totalGrade = bf.readLine();
            Student stu = new Student();
            stu.setName(name);
            stu.setAddress(address);
            stu.setDepartment(department);
            stu.setCreditsEarned(credits);
            stu.setTotalGrade(totalGrade);
            System.out.println("Student details saved");
            session.save(stu);
            break;
        default:
            System.out.println("Exit");
            System.exit(0);
            break;
        }
        session.getTransaction().commit();
        session.close();
        sf.close();

    }
}
```