

## Assignment-5

### Object Oriented Programming Lab

#### Java: Inheritance, Polymorphism, Abstract Class, and Interface

**Name: Sandeep Kumar Singh**

**Reg. No: 2021CA094**

1. You are required to write a base class Student having details (name, registration number, father's name, address (current address, permanent address), and contact details (phone number, email Id). UGStudent, PGStudent, and PhDStudent extend the Student class they have details about the past qualifying degree and marks. Make each attribute protected and provide suitable getter and setter to access attributes.

```
import java.util.Scanner;
class student{
String name;
    String reg;
    String fname;
    String c_add;
    String p_add;
    String phone;
    String email;
}
class UGStudent extends student{
    String degree;
String marks;    void
setter(){
    Scanner sc=new Scanner(System.in);
System.out.print("Enter Name : ");    name
= sc.nextLine();
```

```

        System.out.print("Enter Reg No : ");
reg = sc.nextLine();
        System.out.print("Enter Father's Name : ");
fname = sc.nextLine();
        System.out.print("Enter Current Address : ");
c_add = sc.nextLine();
        System.out.print("Enter Parmanent Address : ");
p_add = sc.nextLine();
        System.out.print("Enter Phone No : ");
phone = sc.nextLine(); System.out.print("Enter
Email Id : ");      email = sc.nextLine();
        System.out.print("Highest Qualify Degree : ");
degree = sc.nextLine();
        System.out.print("Enter Your Marks : ");
marks = sc.nextLine();
    }
    void getter(){
        System.out.println("Name : "+name);
        System.out.println("Reg No : "+reg);
        System.out.println("Father's Name : "+fname);
        System.out.println("Current Address : "+c_add);
        System.out.println("Parmanent Address : "+p_add);
        System.out.println("Phone No : "+phone);
        System.out.println("Email Id : "+email);
        System.out.println("Highest Qualify Degree : "+degree);
        System.out.println("Your Marks : "+marks);
    }
}
class PGStudent extends student{
    String degree;
    String marks;    void
setter(){
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter Name : ");      name
        = sc.nextLine();
        System.out.print("Enter Reg No : ");
reg = sc.nextLine();

```

```

        System.out.print("Enter Father's Name : ");
fname = sc.nextLine();
        System.out.print("Enter Current Address : ");
c_add = sc.nextLine();
        System.out.print("Enter Parmanent Address : ");
p_add = sc.nextLine();
        System.out.print("Enter Phone No : ");
phone = sc.nextLine();
        System.out.print("Enter Email Id : ");
email = sc.nextLine();
        System.out.print("Highest Qualify Degree : ");
degree = sc.nextLine();
        System.out.print("Enter Your Marks : ");
marks = sc.nextLine();
    }
    void getter(){
        System.out.println("Name : "+name);
        System.out.println("Reg No : "+reg);
        System.out.println("Father's Name : "+fname);
        System.out.println("Current Address : "+c_add);
        System.out.println("Parmanent Address : "+p_add);
        System.out.println("Phone No : "+phone);
        System.out.println("Email Id : "+email);
        System.out.println("Highest Qualify Degree : "+degree);
        System.out.println("Your Marks : "+marks);
    }
}
class PHDStudent extends student{
    String degree;
    String marks;    void
    setter(){
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter Name : ");    name
        = sc.nextLine();
        System.out.print("Enter Reg No : ");
        reg = sc.nextLine();
    }
}

```

```

        System.out.print("Enter Father's Name : ");
fname = sc.nextLine();
        System.out.print("Enter Current Address : ");
c_add = sc.nextLine();
        System.out.print("Enter Parmanent Address : ");
p_add = sc.nextLine();
        System.out.print("Enter Phone No : ");
phone = sc.nextLine();
        System.out.print("Enter Email Id : ");
email = sc.nextLine();
        System.out.print("Highest Qualify Degree : ");
degree = sc.nextLine();
        System.out.print("Enter Your Marks : ");
marks = sc.nextLine();
    }
    void getter(){
        System.out.println("Name : "+name);
        System.out.println("Reg No : "+reg);
        System.out.println("Father's Name : "+fname);
        System.out.println("Current Address : "+c_add);
        System.out.println("Parmanent Address : "+p_add);
        System.out.println("Phone No : "+phone);
        System.out.println("Email Id : "+email);
        System.out.println("Highest Qualify Degree : "+degree);
        System.out.println("Your Marks : "+marks);
    }
}

public class assg_5 {    public static void
main(String[] args) {
    UGStudent obj = new UGStudent();
    System.out.println("Enter All Detail of UGStudent Object");
    System.out.println("-----");    obj.setter();

    System.out.println("All Detail of Object");
    System.out.println("-----");    obj.getter();

    PGStudent obj1 = new PGStudent();

```

```

        System.out.println("Enter All Detail of PGStudent Object");
        System.out.println("-----");    obj1.setter();

        System.out.println("All Detail of Object");
        System.out.println("-----");    obj1.getter();

        PHDStudent obj2 = new PHDStudent();
        System.out.println("Enter All Detail of PHDStudent Object");
        System.out.println("-----");    obj2.setter();

        System.out.println("All Detail of Object");
        System.out.println("-----");    obj2.getter();
    }
}

```

### Output—

```

Enter All Detail of UGStudent Object
-----
Enter Name : Ram
Enter Reg No : 2021CA000
Enter Father's Name : Shyam
Enter Current Address : UP
Enter Parmanent Address : Bihar
Enter Phone No : 237887652
Enter Email Id : abc@gmail.com
Highest Qualify Degree : PG
Enter Your Marks : 87
All Detail of Object
-----
Name : Ram
Reg No : 2021CA000
Father's Name : Shyam
Current Address : UP
Parmanent Address : Bihar
Phone No : 237887652
Email Id : abc@gmail.com
Highest Qualify Degree : PG
Your Marks : 87

```

**2. Write another base class Course which stores details about the courses registered by a Student. Course has attributes such as code, name, and credit. UGCourse and PGCourse extends the Course class. The course code follows the following syntax PG-xxxx and UG-xxxx here x is a number which uniquely identifies a course. Make each attribute protected and provide suitable getter and setter to access attributes.**

**Code—**

```
import java.util.Scanner;
class course{
String code;
    String name;
    String credit;
}
class UGCourse extends
course{    String c_id;
void setter(){
    Scanner sc=new Scanner(System.in);
System.out.print("Enter Course Code : ");    c_id
= sc.nextLine();
    System.out.print("Enter Subject Code : ");
code = sc.nextLine();
    System.out.print("Enter Your Name : ");
name = sc.nextLine();    System.out.print("Enter
Credit of Course : ");    credit = sc.nextLine();
}
void getter(){
    System.out.println("Course Code : "+c_id);
    System.out.println("Subject Code : "+code);
    System.out.println("Your Name : "+name);
    System.out.println("Credit of Course : "+credit);
}
}
class PGCourse extends
course{    String c_id;
void setter(){
    Scanner sc=new Scanner(System.in);
System.out.print("Enter Course Code : ");    c_id
= sc.nextLine();
    System.out.print("Enter Subject Code : ");
code = sc.nextLine();
    System.out.print("Enter Your Name : ");
name = sc.nextLine();
```

```

        System.out.print("Enter Credit of Course : ");
        credit = sc.nextLine();
    }
    void getter(){
        System.out.println("Course Code : "+c_id);
        System.out.println("Subject Code : "+code);
        System.out.println("Your Name : "+name);
        System.out.println("Credit of Course : "+credit);
    }
}

public class assg_two_5 {    public static
void main(String[] args) {
    UGCourse o1 = new UGCourse();
    System.out.println("Enter All Detail of UGCouse Object");
    System.out.println("-----");    o1.setter();

    System.out.println("\nAll Detail of UGCourse Object");
    System.out.println("-----");    o1.getter();
    PGCourse o2 = new PGCourse();
    System.out.println("\nEnter All Detail of PGCouse Object");
    System.out.println("-----");    o1.setter();

    System.out.println("\nAll Detail of PGCourse Object");
    System.out.println("-----");    o1.getter();
}
}

```

### Output—

```
Enter All Detail of UGCouse Object
```

```
-----  
Enter Course Code : 121  
Enter Subject Code : 12021  
Enter Your Name : DSA  
Enter Credit of Course : 5
```

```
All Detail of UGCourse Object
```

```
-----  
Course Code : 121  
Subject Code : 12021  
Your Name : DSA  
Credit of Course : 5
```

```
Enter All Detail of PGCourse Object
```

```
-----  
Enter Course Code : 231  
Enter Subject Code : 21031  
Enter Your Name : OPPS  
Enter Credit of Course : 4
```

```
All Detail of PGCourse Object
```

```
-----  
Course Code : 231  
Subject Code : 21031  
Your Name : OPPS  
Credit of Course : 4
```

**3. Write another base class Payment which stores the payment details of the registered Student. Payment has a single attribute paymentID which is unique for each payment. PaymentDD, PaymentUPI, and PaymentNB extends the Payment class, these classes stores details of payment using Demand Draft, UPI, and Net Banking. Add relevant attributes to these classes to store payment details. Make each attribute protected and provide suitable getter and setter to access attributes.**

**Code— import**

```
java.util.Scanner; class
```

```
payment{    String
```

```
name;
```

```
    String regNo;
```

```
    String payMethod;
```



```

    String remarks;
}
class PaymentDD extends
payment{    String
DD_referenceNo;    void
setter(){
    Scanner sc=new Scanner(System.in);
    System.out.print("Enter Your Good Name : ");
    name = sc.nextLine();
    System.out.print("Enter Your Reg No : ");
    regNo = sc.nextLine();
    System.out.print("Enter Payment Method : ");
    payMethod = sc.nextLine();
    System.out.print("Payment DD UTR No : ");
    DD_referenceNo = sc.nextLine();
    System.out.print("Remarks : ");    remarks
    = sc.nextLine();
}
    void getter(){
        System.out.println("Your Good Name : "+name);
        System.out.println("Your Reg No : "+regNo);
        System.out.println("Payment Method : "+payMethod);
        System.out.println("Payment DD UTR No : "+DD_referenceNo);
        System.out.println("Remarks : "+remarks);
    }
}
}
class PaymentUPI extends
payment{    String
UPI_referenceNo;    void
setter(){
    Scanner sc=new Scanner(System.in);
    System.out.print("Enter Your Good Name : ");
    name = sc.nextLine();
    System.out.print("Enter Your Reg No : ");
    regNo = sc.nextLine();
    System.out.print("Enter Payment Method : ");
    payMethod = sc.nextLine();

```

```

        System.out.print("Payment UPI UTR No : ");
        UPI_refrenceNo = sc.nextLine();
        System.out.print("Remarks : ");    remarks
        = sc.nextLine();
    }
    void getter(){
        System.out.println("Your Good Name : "+name);
        System.out.println("Your Reg No : "+regNo);
        System.out.println("Payment Method : "+payMethod);
        System.out.println("Payment UPI UTR No : "+UPI_refrenceNo);
        System.out.println("Remarks : "+remarks);
    }
}

class PaymentNB extends
payment{    String
NB_refrenceNo;    void
setter(){
    Scanner sc=new Scanner(System.in);
    System.out.print("Enter Your Good Name : ");
    name = sc.nextLine();
    System.out.print("Enter Your Reg No : ");
    regNo = sc.nextLine();
    System.out.print("Enter Payment Method : ");
    payMethod = sc.nextLine();
    System.out.print("Payment NB UTR No : ");
    NB_refrenceNo = sc.nextLine();
    System.out.print("Remarks : ");    remarks
    = sc.nextLine();
}
    void getter(){
        System.out.println("Your Good Name : "+name);
        System.out.println("Your Reg No : "+regNo);
        System.out.println("Payment Method : "+payMethod);
        System.out.println("Payment NB UTR No : "+NB_refrenceNo);
        System.out.println("Remarks : "+remarks);
    }
}
}

```

```

public class assg_three_5 {    public static
void main(String[] args) {
    //Payment By DD
    PaymentDD o1 = new PaymentDD();
    System.out.println("Enter All Detail of Payment by DD");
    System.out.println("-----");    o1.setter();

    System.out.println("\nAll Detail of Payment by DD");
    System.out.println("-----");    o1.getter();

    //Payment By UPI
    PaymentUPI o2 = new PaymentUPI();
    System.out.println("\nEnter All Detail of Payment by UPI");
    System.out.println("-----");    o2.setter();

    System.out.println("\nAll Detail of Payment by UPI");
    System.out.println("-----");    o2.getter();

    //Payment By NB
    PaymentNB o3 = new PaymentNB();
    System.out.println("\nEnter All Detail of Payment by NB");
    System.out.println("-----");    o3.setter();

    System.out.println("\nAll Detail of Payment by NB");
    System.out.println("-----");    o3.getter();

    }
}

```

## Code—

Enter All Detail of Payment by DD

---

Enter Your Good Name : Ram  
Enter Your Reg No : 2021CA000  
Enter Payment Method : Bank Draf  
Payment DD UTR No : 3456786  
Remarks : fee paid

All Detail of Payment by DD

---

Your Good Name : Ram  
Your Reg No : 2021CA000  
Payment Method : Bank Draf  
Payment DD UTR No : 3456786  
Remarks : fee paid

Enter All Detail of Payment by UPI

---

Enter Your Good Name : Shyam  
Enter Your Reg No : 2021CA200  
Enter Payment Method : UPI  
Payment UPI UTR No : 3456789  
Remarks : Hostel fee

All Detail of Payment by UPI

---

Your Good Name : Shyam  
Your Reg No : 2021CA200  
Payment Method : UPI  
Payment UPI UTR No : 3456789  
Remarks : Hostel fee

Enter All Detail of Payment by NB

---

Enter Your Good Name : Rahul  
Enter Your Reg No : 2021CA199  
Enter Payment Method : Net banking  
Payment NB UTR No : 3456789  
Remarks : mess fee

All Detail of Payment by NB

---

Your Good Name : Rahul  
Your Reg No : 2021CA199  
Payment Method : Net banking  
Payment NB UTR No : 3456789  
Remarks : mess fee

**4. Add one instance of each Course class and Payment class to the student class so that course and payment details of a student can be associated with each student. Add a member function in Student class to print course detail and payment details using student's**

registration number. Demonstrate the program by creating 10 students and printing their details.

Code—

```
import java.util.Scanner;
class Student{
String name;
    String regNo;

    void set() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Student Detail :-");
        System.out.print("Student Name : ");    name
= sc.nextLine();
        System.out.print("Student Reg No : ");
regNo = sc.nextLine();
    }
    void Display() {
        System.out.println("Student Name : "+name);
        System.out.println("Student Reg No : "+regNo);
    }
}
class Course extends Student{
    String courseName;

    Course() {
        Scanner sc = new Scanner(System.in);
set();
        System.out.print("Which Course Buy : ");
courseName = sc.nextLine();
    }
    void get() {
        System.out.println("\nAfter Course obj Calling--");
super.Display();
        System.out.println("Course Buy : "+courseName);
    }
}
```

```

class Payment extends Student{
    String paymentDetail;

    Payment() {
        set();
        Scanner sc = new Scanner(System.in);
        System.out.print("Method of Payment : ");
        paymentDetail = sc.nextLine();
    }
    void get() {
        System.out.println("\nAfter Payment obj Calling--");
        super.Display();
        System.out.println("Payment Method : "+paymentDetail);
    }
}

public class assg_four_5 {    public static void
main(String[] args) {
    System.out.println("Student Detail:--");
        System.out.println("-----");
        Course c = new Course();
        Payment p = new Payment();

        c.get();
        p.get();
    }
}

```

## Output—

```
Student Detail:--
-----
Student Detail :-
Student Name : Ram
Student Reg No : 2021CA000
Which Course Buy : Udemy
Student Detail :-
Student Name : Ram
Student Reg No : 2021CA000
Method of Payment : Upi

After Course obj Calling--
Student Name : Ram
Student Reg No : 2021CA000
Course Buy : Udemy

After Payment obj Calling--
Student Name : Ram
Student Reg No : 2021CA000
Payment Method : Upi
```

**5. Write a class Employee having attributes: employeeID, name, department, dob, designation, yearOfJoining, and phoneNumber. Add a function to print details of employee using employeeID. Two classes, i.e., Faculty and OfficeStaff, extends the Employee class. The Faculty has a list of subjects and labs taught by them. OfficeStaff has a list of skills which stores skill such as Typing, Technician, etc. Each of the above classes has appropriate getter setter for inserting skill in the list and printing the list of skill. Demonstrate above class using a Test class which creates 5 employees of different types and print their details.**

## Code—

```
import java.util.Scanner;

class Employee {
    int emplId;
    String name, dept, des, yoj, phone;

    void print() {
        System.out.println("Employee Id : " + emplId);
        System.out.println("Employee Name : " + name);
        System.out.println("Employee Department : " + dept);
    }
}
```

```

        System.out.println("Employee Destination : " + des);
        System.out.println("Employee Year of Join : " + yoj);
        System.out.println("Employee Phone Number : " + phone);
    }
}

class Faculty extends Employee {
    String sub;

    void setter() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Employee Id : ");    emplId
        = sc.nextInt();
        System.out.print("Enter Employee Name : ");
        name = sc.nextLine();
        System.out.print("Enter Employee Department : ");
        dept = sc.nextLine();
        System.out.print("Enter Employee Destination : ");
        des = sc.nextLine();
        System.out.print("Enter Employee Year of Join : ");
        yoj = sc.nextLine();
        System.out.print("Enter Employee Phone : ");
        phone = sc.nextLine();
        System.out.print("Enter Sub List : ");
        sub = sc.nextLine();
    }
    void getter() {
        print();
        System.out.println("Subject List : " + sub);
    }
}

class OfficeStaff extends Employee {
    String skill;

    void setter() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Employee Id : ");    emplId
        = sc.nextInt();

```



```

        System.out.print("Enter Employee Name : ");
name = sc.nextLine();
        System.out.print("Enter Employee Department : ");
dept = sc.nextLine();
        System.out.print("Enter Employee Destination : ");
des = sc.nextLine();
        System.out.print("Enter Employee Year of Join : ");
yoj = sc.nextLine();
        System.out.print("Enter Employee Phone : ");
phone = sc.nextLine();
        System.out.print("Enter Employee Skill : ");
skill = sc.nextLine();
    }    void
getter() {
print();
        System.out.println("Employee Skill : "+skill);
    }
}
public class assg_five_5 {
    public static void main(String[] args) {
        Faculty f = new Faculty();
        f.setter();
        System.out.println("\nEmployee Detail");
        System.out.println("-----");
        f.getter();

        System.out.println();
        OfficeStaff o = new OfficeStaff();
        o.setter();

        System.out.println("\nEmployee Detail");
        System.out.println("-----");
        o.getter();
    }
}

```

## Output—

```
Enter Employee Id : 2
Enter Employee Name : Enter Employee Department : Ram CSE
Enter Employee Destination : SDE
Enter Employee Year of Join : 20218
Enter Employee Phone : 4356789567
Enter Sub List : C/C++ JAVA
```

### Employee Detail

```
-----
Employee Id : 2
Employee Name :
Employee Department : Ram CSE
Employee Destination : SDE
Employee Year of Join : 20218
Employee Phone Number : 4356789567
Subject List : C/C++ JAVA
Enter Employee Id : 2
Enter Employee Name : Enter Employee Department : Ram CSE
Enter Employee Destination : SDE
Enter Employee Year of Join : 2018
Enter Employee Phone : 234567898
Enter Sub List : C/C++ Java
Enter Employee Skill : Technician
```

### Employee Detail

```
-----
Employee Id : 3
Employee Name :
Employee Department : Shyam Civil
Employee Destination : Maruti
Employee Year of Join : 2022
Employee Phone Number : 456789
Employee Skill : Technician
```

**6. Classes HOD, DUGC, DMPC, and DDPC extends the Faculty class (given in the above question). These classes stores detail specific to the Head of the Department (HOD), Department Undergraduate Committee (DUGC) Convenor, Department Master Program Committee (DUGC)Convenor, and Department Doctoral Program Committee (DUGC) Convenor. SkilledStaff and UnSkilledStaff extends class OfficeStaff (given in the above question). Each class has a public method toString( ) which returns their details as a String. Demonstrate above class using a Test class which creates 6 employees of different types and print their details.**

## Code—

```
import java.util.Scanner;
```

```

class Employee {
    int emplId;
    String name, dept, des, yoj, phone;

    void print() {
        System.out.println("Employee Id : " + emplId);
        System.out.println("Employee Name : " + name);
        System.out.println("Employee Department : " + dept);
        System.out.println("Employee Destination : " + des);
        System.out.println("Employee Year of Join : " + yoj);
        System.out.println("Employee Phone Number : " + phone);
    }
}

class Faculty extends Employee {
    String sub;

    void setter() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Employee Id : ");    emplId
        = sc.nextInt();
        System.out.print("Enter Employee Name : ");
        name = sc.nextLine();
        System.out.print("Enter Employee Department : ");
        dept = sc.nextLine();
        System.out.print("Enter Employee Destination : ");
        des = sc.nextLine();
        System.out.print("Enter Employee Year of Join : ");
        yoj = sc.nextLine();
        System.out.print("Enter Employee Phone : ");
        phone = sc.nextLine();
        System.out.print("Enter Sub List : ");
        sub = sc.nextLine();
    }

    void getter() {
        print();
        System.out.println("Subject List : " + sub);
    }
}

```

```

    }
}
class OfficeStaff extends Employee {
    String skill;

    void setter() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Employee Id : ");    emplId
        = sc.nextInt();
        System.out.print("Enter Employee Name : ");
        name = sc.nextLine();
        System.out.print("Enter Employee Department : ");
        dept = sc.nextLine();
        System.out.print("Enter Employee Destination : ");
        des = sc.nextLine();
        System.out.print("Enter Employee Year of Join : ");
        yoj = sc.nextLine();
        System.out.print("Enter Employee Phone : ");
        phone = sc.nextLine();
    }
    void getter() {
        print();
        System.out.println("Employee Skill : " + skill);
    }
}
class Skilled extends OfficeStaff {
    void setter() {
        Scanner sc = new Scanner(System.in);
        super.setter();
        System.out.print("Enter Employee Skill : ");
        skill = sc.nextLine();
    }
    void getter() {
        super.getter();
    }
}
class UnSkilled extends OfficeStaff {
    void setter() {

```

```

        Scanner sc = new Scanner(System.in);
super.setter();
    }    void getter() {
super.getter();
    }
}
class HOD extends Faculty {
void setter() {
super.setter();
    }    void getter() {
super.getter();
    }
}
class DUGC extends Faculty {
void setter() {
super.setter();
    }    void getter() {
super.getter();
    }
}
class DMGC extends Faculty {
    void setter() { super.setter();
    }

    void getter() {
super.getter();
    }
}
class DDPC extends Faculty {
void setter() {
super.setter();
    }    void getter() {
super.getter();
    }
}
public class assg_six_5 {

```

```

    public static void main(String[] args) {
        HOD f = new HOD();
        f.setter();

        System.out.println("\nEmployee Detail");
        System.out.println("-----");
        f.getter();

        System.out.println();
        Skilled o = new Skilled();
        o.setter();

        System.out.println("\nEmployee Detail");
        System.out.println("-----");
        o.getter();

        System.out.println();
        UnSkilled o1 = new UnSkilled();
        o1.setter();

        System.out.println("\nEmployee Detail");
        System.out.println("-----");    o1.getter();
    }
}

```

## 7. Employee and ProductionWorker Classes

Design a class named Employee . The class should keep the following information in fields:

- Employee name

Employee number in the format XXX–L, where each X is a digit within the range 0–9 and the L is a letter within the range A–M.

- Hire date

Write one or more constructors and the appropriate accessor and mutator methods for the class. Next, write a class named ProductionWorker that extends the Employee class. The

**ProductionWorker** class should have fields to hold the following information:

- Shift (an integer)
- Hourly pay rate (a double )

The workday is divided into two shifts: day and night. The shift field will be an integer value representing the shift that the employee works. The day shift is shift 1 and the night shift is shift 2. Write one or more constructors and the appropriate accessor and mutator methods for the class. Demonstrate the classes by writing a program that uses a **ProductionWorker** object.

Code—

```
import java.util.Scanner;
```

```
class Employee {  
    String empName;  
    String empld;  
    String hireDate;
```

```
    Employee() {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter Employee Name : ");  
        empName = sc.nextLine();  
        System.out.print("Enter Employee Id(XXX-L) : ");  
        empld = sc.nextLine();  
        System.out.print("Enter Employee HireDate : ");  
        hireDate = sc.nextLine(); }  
}
```

```
    void accessor() {  
        System.out.println("Employee Id : "+empld);  
        System.out.println("Employee Name : "+empName);  
        System.out.println("Employee Hiredate : "+hireDate);  
    }  
}
```

```

class ProductionWorker extends
Employee {    int shift;    double
hourPayrate;

    ProductionWorker() {
super();
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Employee Shift(1 for day and 2 for
night) : ");    shift = sc.nextInt();
        System.out.print("Enter Employee Hourly Pay Rate : ");
hourPayrate = sc.nextDouble();
    }
    void accessor() {
super.accessor();    if(shift ==
1)
        System.out.println("Employee Shift : Day");
else
        System.out.println("Employee Shift : Nigth");
        System.out.println("Hourly Pay Rate : "+hourPayrate);
    }
}

public class assig_seven_5 {
    public static void main(String[] args) {
        ProductionWorker p = new ProductionWorker();

        System.out.println();
        System.out.println("Employee Detail--");
        System.out.println("-----");
        p.accessor();
    }
}

```



### Output—

```
Enter Employee Name : Ram
Enter Employee Id(XXX-L) : 123C
Enter Employee HireDate : 12/10/2012
Enter Employee Shift(1 for day and 2 for night) : 2
Enter Employee Hourly Pay Rate : 45
```

Employee Detail--

```
-----
Employee Id : 123C
Employee Name : Ram
Employee Hiredate : 12/10/2012
Employee Shift : Nigth
Hourly Pay Rate : 45.0
```

### 8. ShiftSupervisor Class

In a particular factory, a shift supervisor is a salaried employee who supervises a shift. In addition to a salary, the shift supervisor earns a yearly bonus when his or her shift meets production goals.

Design a ShiftSupervisor class that extends the Employee class you created in question 7. The ShiftSupervisor class should have a field that holds the annual salary and a field that holds the annual production bonus that a shift supervisor has earned. Write one or more constructors and the appropriate accessor and mutator methods for the class. Demonstrate the class by writing a program that uses a ShiftSupervisor object.

### Code—

```
import java.util.Scanner;
```

```
class Employee {
    String empName;
    String empId;
    String hireDate;
```

```
    Employee() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Employee Name : ");
        empName = sc.nextLine();
        System.out.print("Enter Employee Id(XXX-L) : ");
        empId = sc.nextLine();
```

```

        System.out.print("Enter Employee HireDate : ");
        hireDate = sc.nextLine();
    }
    void accessor() {
        System.out.println("Employee Id : "+empld);
        System.out.println("Employee Name : "+empName);
        System.out.println("Employee Hiredate : "+hireDate);
    }
}

class ProductionWorker extends
Employee {    int shift;    double
hourPayrate;

    ProductionWorker() {
super();
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Employee Shift(1 for day and 2 for
night) : ");    shift = sc.nextInt();
        System.out.print("Enter Employee Hourly Pay Rate : ");
        hourPayrate = sc.nextDouble();
    }
    void accessor() {
super.accessor();    if
(shift == 1)
        System.out.println("Employee Shift : Day");
    else
        System.out.println("Employee Shift : Nigth");

        System.out.println("Hourly Pay Rate : " + hourPayrate);
    }
}

class ShiftSupervisor extends
Employee {    int anualSalry;    int
anualProdBonus;

    ShiftSupervisor() {
super();

```

```

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Annual Salary : ");
        anualSalry = sc.nextInt();    System.out.print("Enter
        Annual Production Bonus : ");    anualProdBonus =
        sc.nextInt();
    }
    void accessor() {
        super.accessor();
        System.out.println("Enter Annual Salary : " + anualSalry);
        System.out.println("Enter Annual Production Bonus : " +
        anualProdBonus);
    }
}
public class assig_seven_5 {    public
static void main(String[] args) {
    ShiftSupervisor s = new ShiftSupervisor();

    System.out.println();
    System.out.println("Employee Detail--");
    System.out.println("-----");
    s.accessor();
}
}

```

### Output—

```

Enter Employee Name : Ram
Enter Employee Id(XXX-L) : 123A
Enter Employee HireDate : 12/08/2010
Enter Annual Salary : 12678
Enter Annual Production Bonus : 1200

```

```

Employee Detail--
-----

```

```

Employee Id : 123A
Employee Name : Ram
Employee Hiredate : 12/08/2010
Enter Annual Salary : 12678
Enter Annual Production Bonus : 1200

```

## 9. TeamLeader Class

In a particular factory, a team leader is an hourly paid production worker that leads a small team. In addition to hourly pay, team leaders earn a fixed monthly bonus. Team leaders are required to

attend a minimum number of hours of training per year. Design a TeamLeader class that extends the ProductionWorker class you designed in question 7. The TeamLeader class should have fields for the monthly bonus amount, the required number of training hours, and the number of training hours that the team leader has attended. Write one or more constructors and the appropriate accessor and mutator methods for the class. Demonstrate the class by writing a program that uses a TeamLeader object.

Code—

```
import java.util.Scanner;

class Employee {
    String empName;
    String empld;
    String hireDate;

    Employee() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Employee Name : ");
        empName = sc.nextLine();
        System.out.print("Enter Employee Id(XXX-L) : ");
        empld = sc.nextLine();
        System.out.print("Enter Employee HireDate : ");
        hireDate = sc.nextLine();
    }

    void accessor() {
        System.out.println("Employee Id : "+empld);
        System.out.println("Employee Name : "+empName);
        System.out.println("Employee Hiredate : "+hireDate);
    }
}

class ProductionWorker extends
Employee {    int shift;    double
hourPayrate;
```

```

    ProductionWorker() {
super();
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Employee Shift(1 for day and 2 for
night) : ");    shift = sc.nextInt();
        System.out.print("Enter Employee Hourly Pay Rate : ");
hourPayrate = sc.nextDouble();    }
        void accessor() {
super.accessor();    if
(shift == 1)
            System.out.println("Employee Shift : Day");
else
            System.out.println("Employee Shift : Nighth");

        System.out.println("Hourly Pay Rate : " + hourPayrate);
    }
}
class TeamLeader extends
ProductionWorker {    int monthlyBonus;
int trainHour;

    TeamLeader() {
super();
        Scanner sc = new Scanner(System.in);
System.out.print("Enter Monthly Bonus : ");
monthlyBonus = sc.nextInt();
        System.out.print("Enter Training Hours : ");
trainHour = sc.nextInt();
    }
        void accessor() {
super.accessor();
            System.out.println("Monthly Bonus : " + monthlyBonus);
            System.out.println("Training Hour : " + trainHour);
        }
    }
}
public class assig_seven_5 {    public
static void main(String[] args) {

```

```

TeamLeader s = new TeamLeader();

System.out.println();
System.out.println("Employee Detail--");
System.out.println("-----");    s.accessor();
}
}

```

### Output—

```

Enter Employee Name : Ram
Enter Employee Id(XXX-L) : 123B
Enter Employee HireDate : 13/02/2000
Enter Employee Shift(1 for day and 2 for night) : 2
Enter Employee Hourly Pay Rate : 56
Enter Monthly Bonus : 120
Enter Training Hours : 8

```

```

Employee Detail--
-----
Employee Id : 123B
Employee Name : Ram
Employee Hiredate : 13/02/2000
Employee Shift : Nighth
Hourly Pay Rate : 56.0
Monthly Bonus : 120
Training Hour : 8

```

10. Repeat Q1 and Q5 using abstract classes.
11. Create an interface Vehicle having three methods changeGear, speedup, and applyBrakes. Two other classes, Bike and Bicycle, implement the Vehicle interface. These classes have two fields, i.e., gear and speed. The changeGear method accepts the value of gear and assigns it to the gear field. The speedup method accepts the value by which the speed field should be increased. The applyBrakes method accepts the value by which the speed field should be decreased. Create a Test class to demonstrate each of these classes.

### Code—

```
import java.io.*;
```

```
interface Vehicle {
```

```
    // all are the abstract methods.
```

```

    void changeGear(int a);
    void speedUp(int a);    void
    applyBrakes(int a);
}
class Bicycle implements Vehicle{
    int
    speed;    int
    gear;

    // to change gear
    @Override
    public void changeGear(int newGear){
        gear =
    newGear;
    }

    // to increase speed
    @Override
    public void speedUp(int increment){

        speed = speed + increment;
    }

    // to decrease speed
    @Override
    public void applyBrakes(int decrement){

        speed = speed - decrement;
    }
    public void printStates() {
        System.out.println("speed: " + speed
            + " gear: " + gear);
    }
}
class Bike implements Vehicle {

```

```

        int
speed;    int
gear;

        // to change gear
@Override
    public void changeGear(int newGear){
        gear =
newGear;
    }

        // to increase speed
@Override
    public void speedUp(int increment){

        speed = speed + increment;
    }

        // to decrease speed
@Override
    public void applyBrakes(int decrement){

        speed = speed - decrement;
    }
    public void printStates() {
        System.out.println("speed: " + speed
            + " gear: " + gear);
    }
}
class assg_eleven_5 {

    public static void main (String[] args) {

        // creating an inatance of Bicycle
        // doing some operations
        Bicycle bicycle = new Bicycle();
        bicycle.changeGear(2);
    }
}

```



```
bicycle.speedUp(3);  
bicycle.applyBrakes(1);
```

```
    System.out.println("Bicycle present state :");  
bicycle.printStates();
```

```
    // creating instance of bike.  
Bike bike = new Bike();  
bike.changeGear(1);  
bike.speedUp(4);  
bike.applyBrakes(3);
```

```
    System.out.println("Bike present state :");  
bike.printStates();  
    }  
}
```

**Output—**

```
Bicycle present state :  
speed: 2 gear: 2  
Bike present state :  
speed: 1 gear: 1
```

**12. Create an interface Shape having two methods getArea and getPerimeter. Three classes, Circle, Triangle, and Rectangle, implement the Shape interface, and override the two methods. Create a Test class to demonstrate each of these classes.**

**Code—**

```
interface Shape  
{  
    void input();  
    void area();  
}  
class Circle implements Shape  
{    int r = 0;    double pi =  
3.14, ar = 0;  
    @Override    public  
void input()
```

```

    {      r = 5;    }
@Override    public
void area()
    {
        ar = pi * r * r;
        System.out.println("Area of circle:"+ar);
    }
}
class Rectangle extends Circle
{
    int l = 0, b = 0;
    double ar;    public
    void input()
    {
        super.input();
    l = 6;      b = 4;    }
    public void area()
    {
        super.area();
    ar = l * b;
        System.out.println("Area of rectangle:"+ar);
    }
}
public class assg_twelve_5
{
    public static void main(String[] args)
    {
        Rectangle obj = new Rectangle();
    obj.input();      obj.area();
    }
}

```

**Output—**

```

Area of circle:78.5
Area of rectangle:24.0

```