USER INPUT

```
System.out.println("Enter a value");
        Scanner in = new Scanner(System.in);
        int num = in.nextInt();
        System.out.println(num);
```

ASSIGNMENT -----> 1

```
class Tester {
 public static void main(String[] args) {
  // Implement your code here
  int num1=6;
  int num2=5;
  int sum=0;
  if(num1==num2)
  {
      sum=num1+num2;
  }
  else
  {
      sum=(2*(num1+num2));
  }
  System.out.println(sum);
 }
}
```

****

ASSIGNMENT ----> 2

```
class Tester {
 public static void main(String[] args) {
  // Implement your code here
  // Implement your code here
  float a=1;
  float b=4;
  float c=6;
  float root1=0;
                float root=0;
  float root2=0;
  float discrimination=0;
  discrimination=((b*b)-(4*a*c));

  if (discrimination==0)
  {
                    root=(-b/(2*a));
      System.out.println(root);
  }
  else if (discrimination>0)
  {
            root1=((-b-discrimination)/(2*a));
  root2=((-b+discrimination)/(2*a));
     System.out.println(root1);
     System.out.println(root2);
  }
  else
  {
     System.out.println("The equation has no real roots");
  }
 }
}
```

****

ASSIGNMENT ----> 3

```
class Tester {
 public static void main(String[] args) {
  // Implement your code here
  int a=2;
  int b=6;
  int c=7;
  int product;
  if ((a!=7)&&(b!=7)&&(c!=7))
  {
     product=a*b*c;
  }
  else if(a==7)
  {
     product=b*c;
  }
  else if(b==7)
  {
     product=c;
  }
  else
  {
     product=(-1);
  }
  System.out.println("Product is: "+product);

 }
}
```

****

ASSIGNMENT ----> 4

```
class Tester {
 public static void main(String[] args) {
  // Implement your code here
  char foodType='V';
  int quantity=1;
  int distance=7;
  int billAmount;
  int foodPrice;
  int delivery;
  if(((foodType=='V')||(foodType=='N'))&&(quantity>=1)&&(distance>0))
  {
      {
      if(foodType=='N')
      {
          foodPrice=(quantity*15);
      }
      else
      {
          foodPrice=(quantity*12);
      }

     }
      {
          if(distance<=3)
          {
              delivery=0;
          }
          else if((distance>3) && (distance<6))
          {
              delivery=((3*0)+((distance-3)*1));
          }
          else
          {
              delivery=(((distance-6)*2)+(3*1));
          }
      }
      billAmount=foodPrice+delivery;
      System.out.println("billAmount: "+billAmount);
    }
  else
  {
      System.out.println("-1");
  }
 }
}
```

****

ASSIGNMENT ----> 5

```
class Tester {
 public static void main(String[] args) {
  // Implement your code here
  int accountNumber=1001;
  int salary=40000;
  int accountBalance=250000;
  String loanType="Car";
  int loanAmountExpected=300000;
  int loanAmount=0;
  int emisExpected=30;
  int emisNum=0;
  if((accountNumber>=1000)&&(accountNumber<2000)&&(accountBalance>=1000))
  {
      if((salary>25000)&&(loanType.equals("Car")))
      {
          loanAmount=500000;
              emisNum=36;
      }
      else if((salary>=50000)&&(salary<75000)&&(loanType.equals("House")))
      {
              loanAmount=6000000;
              emisNum=60;
      }
      else if((salary>=75000)&&(loanType.equals("Business")))
      {
              loanAmount=75000000;
              emisNum=86;
      }
      if((loanAmountExpected<=loanAmount)&&(emisExpected<=emisNum))
      {

              System.out.println("eligible loanAmount="+loanAmount);
              System.out.println("eligible emisNum="+emisNum);
      }

      else
      {
              System.out.println("Bank doesn't provide loan");
      }
      }

  else
  {
      if((accountNumber<=1000)&&(accountNumber>2000))
      {
      System.out.println("Account number doesn't match");
      }
      else
      {
          System.out.println("Balance doesn't reach the criteria");
      }
  }
 }
}
```

****

ASSIGNMENT ----> 6

```
class Tester {
 public static void main(String[] args) {
  // Implement your code here
  int oneDollar=2;
  int fiveDollar=4;
  int purchaseAmount=21;
  int fives;
  int oneNotes;
  int fiveNotes;
  if(purchaseAmount<=((1*oneDollar)+(5*fiveDollar)))
  {
    fives=fiveDollar*5;
    oneNotes=purchaseAmount-fives;
    fiveNotes=(fives/5);
    if(oneNotes>0&&fives>0)
    {
    System.out.println("$1 notes needed = "+oneNotes);
    System.out.println("$5 notes needed = "+fiveNotes);
    }
    else
    {
        System.out.println("-1");
    }}
      else{
        System.out.println("-1");
  }
 }
}
```

****

ASSIGNMENT -----> 7

```
class Tester {
 public static void main(String[] args) {
  // Implement your code here
  int day=29;
  int month=2;
  int year=16;
```

```
String a="20";
if (month==4||month==6||month==9||month==11)
{
    if(day==30)
    {
        day=1;
        month=month+1;
    }
    else{
        day=day+1;
    }
    System.out.println(day+"-"+month+"-"+"20"+year);
}
else if(month==2 && day<=28)
{
    if(day==28 && year%4==0)
    {
        day=day+1;

    }
    else{
        day=day+1;
    }
     System.out.println(day+"-"+month+"-"+"20"+year);
}
else if(month==2 && day<=29)
{
     if(day==29)
    {
        day=1;
        month=month+1;
    }
    else{
        day=day+1;
    }
     System.out.println(day+"-"+month+"-"+"20"+year);
}
else if(month==1||month==3||month==5||month==7||month==10) {
    if(day==31)
    {
        day=1;
        month=month+1;
    }
    else{
        day=day+1;
    }
    System.out.println(day+"-"+month+"-"+"20"+year);

}
else if(month==12)
{
    if(day==31)
    {
        day=1;
```

```
         month=1;
         year=year+1;
      }
      else{
         day=day+1;
         System.out.println(day+"-"+month+"-"+"20"+year);
      }
  }
 }
}
```

****

ASSIGNMENT -----> 8

```
class Tester {
 public static void main(String[] args) {
  // Implement your code here
  int num1=35;
     if((num1%3==0)&&(num1%5==0))
     {
        System.out.println("Zoom");
     }
     else if (num1%5==0)
     {
        System.out.println("Zap");
     }
     else if (num1%3==0)
     {
        System.out.println("Zip");
     }
     else
     {
        System.out.println("Invalid");
     }
 }
}
```

****

ITERATION CONTROL STRUCTURES

ASSIGNMENT ----> 1

```
class Tester {
 public static void main(String[] args) {
  // Implement your code here
  int org=1331;
  int rev=0;
  int remainder;
  int num1=org;
     while (num1>0)
     {
        remainder=num1%10;
        rev=(rev*10)+remainder;
        num1=num1/10;
     }
     if(org==rev)
     {
        System.out.println(rev+" is a palindrome");
     }
     else{
        System.out.println(rev+" is not a palindrome");
     }

 }
}
```

****

ASSIGNMENT -----> 2

```
class Tester {
 public static void main(String[] args) {
  // Implement your code here
  int heads=8;
  int legs=12;
  int rabbit=0, chickens=0;

      rabbit=((legs/2)-(heads));
      chickens=heads-rabbit;
      if(rabbit>0&&chickens>0&&legs%2==0)
      {
      System.out.println("Chickens="+chickens);
      System.out.println("Rabbits="+rabbit);
   }
  else{
      System.out.println("The number of chickens and rabit cannot be found");
   }

 }
}
```

****
***

ASSIGNMENT -----> 3

```
class Tester {
 public static void main(String[] args) {
  // Implement your code here
  int num=2250;
  int remainder=0;
  int sum=0;
  int num1=num;
  while(num1>0)
  {
     remainder=num1%10;         rem=2250%10=0    225%10=5   22%10=2  2%10=2
     sum=sum+remainder;          0+0             0+5        5+2=7    7+2=9; sum=9
     num1=num1/10;            2250/10=225     225/10=22  22/10=2  2/10=0
  }
  if((num%sum)==0)              2250%9==0
  {
     System.out.println(num+" is divisible by sum of its digits");
  }
  else
  {
     System.out.println(num+" is not divisible by sum of its digits");
  }
 }
}
```

****

ASSIGNMENT -----> 4

```
class Tester {
 public static void main(String[] args) {
  // Implement your code here
  int num=123;
  int num2=738;
  int multi=num;
  int remainder;
  int num1=num;
  while(num1>0)
  {
     remainder=num1%10;
     multi=multi*remainder;
     num1=num1/10;
  }

  if(multi==num2)
  {

    System.out.println(num+" is a seed of "+multi);
  }
  else{
      System.out.println(num+" is not a seed of "+multi);
  }
 }
}
```

****

ASSIGNMENT -----> 5

```
class Tester {
 public static void main(String[] args) {
  // Implement your code here

  int num=371;
  int remainder;
  int num1=num;
  double cubes=0;
  double sum=0;
  while(num1>0)
  {
      remainder=num1%10;
      cubes=Math.pow(remainder,3);
      sum=sum+cubes;
      num1=num1/10;
  }
  if(sum==num)
  {
       System.out.println(num+" is an Armstrong number");
  }
  else{
       System.out.println(num+" is not an Armstrong number");
  }
 }
}
```

****

ASSIGNMENT -----> 6

```
class Tester {
 public static void main(String[] args) {
  // Implement your code here
  int num=15;
  double remainder;
  double rem=0;
  double squares=0;
  double rmdr=0;
  double sum=0;
  int num1=num;
  while(num1>0) {
      remainder=(num1%10);
      squares=Math.pow(remainder,2);
      rem=squares+rem;
      num1=num1/100;
      if(num1==1) {
          num1=num/10;
          rmdr=num1%10;
          squares=Math.pow(rmdr,2);
          rem=squares;
      }}
  if(rem%9==0) {
       System.out.println("The number "+num+" is a lucky number");
  }
  else{
      System.out.println("The number "+num+" is not a lucky number");
  }
 }
}
```

****
**

ASSIGNMENT -----> 7

```
class Tester {
 public static void main(String[] args) {
  // Implement your code here
  int num1=5;
  int num2=10;
  int grt=0;
  if(num1>num2)
  {
      grt=num1;
  }
  else{
      grt=num2;
  }
  int num=grt;
  while(grt%num1!=0)
  {
        grt=grt+num;
  }
   System.out.println(grt);
     }
}
```

****

ASSIGNMENT ----->   8

```
class Tester {
 public static void main(String[] args) {
  // Implement your code here
  int rows = 5;

    for (int i = rows; i >= 1; i--)
    {
      for (int j = 1; j <= i; j++)
      {
        System.out.print("*");
      }
      System.out.println();
    }


 }
}
```

****

```
class Tester {
 public static void main(String[] args) {
  // Implement your code here
  int num=1950;
  double remainder;
  double rem=0;
  double squares=0;
  double sum=0;
  int num1=num;
  while(num1>1)
  {
      remainder=(num1%10);
      squares=Math.pow(remainder,2);
      rem=squares+rem;
      num1=num1/100;

  }
    System.out.println(rem);
  if(rem%9==0)
  {
      System.out.println("The number "+num+" is a lucky number");
  }
  else{
      System.out.println("The number "+num+" is not a lucky number");
  }
 }
}
```

METHODS----->>

ASSIGNMENT ----->1 (INCOMPLETE)

ASSIGNMENT ------>> 3

```
class Calculator {
public int num;
public int sumOfDigits()
{
    int sum=0;
    int num1=num;
    while(num1>0) {
    int rem=num1%10;
    sum=sum+rem;
    num1=num1/10;
    }
    return sum;
}   }
class Tester {
 public static void main(String args[]) {
  Calculator calculator = new Calculator();
  System.out.println(calculator.sumOfDigits());
 }
}
```

****

ASSIGNMENT ----> 4

```
class Rectangle {
    public float length;
    public float width;
    public double calculateArea()
    {
        double Area=(length*width);
        return (Math.round(Area*100.0)/100.00);
    }

    public double calculatePerimeter()
    {
    double Perimeter=(2*(length+width));
```

```
        return (Math.round(Perimeter*100.0)/100.00);
    }
        }
class Tester {

 public static void main(String args[]) {

  Rectangle rectangle=new Rectangle();
  System.out.println("Area of the rectangle is "+rectangle.calculateArea());
  System.out.println("Perimeter of the rectangle is "+rectangle.calculatePerimeter());
 }

}
```

****
****

ASSIGNMENT ----> 2 (SAI SOLVED)

```
//constructor and this keyword - assignment2
class Restaurant{
    public String name;
    public long restaurantContact;
    public String restaurantAddress;
    public float rating;

    Restaurant(String name,long restaurantContact,String restaurantAddress,float rating){
        this.name=name;
        this.restaurantContact=restaurantContact;
        this.restaurantAddress=restaurantAddress;
        this.rating=rating;
    }

    public void displayRestaurantDetails(){
        System.out.println("Restaurant Name : "+this.name);
        System.out.println("Restaurant Contact : "+ this.restaurantContact);
    }
}

class Tester{
    public static void main(String[] args){
        Restaurant restaurant1=new Restaurant("Jalpan",6738892347L,"Mysore",4.3f);
        restaurant1.displayRestaurantDetails();
    }
}
```

****
*

ENCAPSULATION ----->

ASSIGNMENT ------>> 1

```
class Employee {

 private String employeeId;
 private String employeeName;
 private int salary;
 private int bonus;
 private int jobLevel;



 public String getEmployeeId() {
     return employeeId;
 }
 public void setEmployeeId(String employeeId) {
     this.employeeId=employeeId;
 }
 public String getEmployeeName() {
     return employeeName;
 }
 public void setEmployeeName(String employeeName) {
     this.employeeName=employeeName;
 }
 public int getSalary() {
     return salary;
 }
 public void setSalary(int salary){
    this.salary=salary;
 }
 public int getJobLevel() {
         return jobLevel;
     }
     public void setJobLevel(int jobLevel) {
         this.jobLevel=jobLevel;
     }
     public int getBonus() {
         return bonus;
     }
     public void setBonus(int bonus) {
         this.bonus=bonus;
     }
 public void calculateSalary() {

  if (this.jobLevel >= 4) {
   this.bonus = 100;
  } else {
   this.bonus = 50;
  }
  this.salary += this.bonus;
 }
}

class Tester {

 public static void main(String args[]) {

  Employee employee = new Employee();
  employee.setEmployeeId ("C101");
  employee.setEmployeeName ("Steve");
  employee.setSalary(650);
  employee.setJobLevel(4);

  employee.calculateSalary();

  System.out.println("Employee Details");
  System.out.println("Employee Id: " + employee.getEmployeeId());
  System.out.println("Employee Name: " + employee.getEmployeeName());
  System.out.println("Salary: " + employee.getSalary());

 }
}
```

****
*****

ASSIGNMENT ----->> 2

```
class MovieTicket {
    //Implement your code here
    private int movieId;
    private int noOfSeats;
    private double costPerTicket;
    public MovieTicket(int movieId,int noOfSeats)
    {
        this.movieId=movieId;
        this.noOfSeats=noOfSeats;
    }
    public int getMovieId()
    {
        return this.movieId;
    }
    public void setMovieId(int movieId)
    {
        this.movieId=movieId;
    }
    public int getNoOfSeats()
    {
        return this.noOfSeats;
    }
    public void setNoOfSeats(int noOfSeats)
    {
        this.noOfSeats=noOfSeats;
    }
    public double getCostPerTicket()
    {
        return this.costPerTicket;
    }
    public void setCostPerTicket(double costPerTicket)
    {
        this.costPerTicket=costPerTicket;
    }
    public double calculateTotalAmount()
    {
        if(movieId==111)
           {
            costPerTicket=7;
           }
        else if(movieId==112)
           {
            costPerTicket=8;
           }
        else if(movieId==113)
           {
            costPerTicket=8.5;
           }
        else
           {
            costPerTicket=0;
           }
        return costPerTicket*noOfSeats*1.02;
    }
}



class Tester {
    public static void main(String[] args) {
        int a=112,b=3;
        MovieTicket movieTicket = new MovieTicket(a,b);

        double amount = movieTicket.calculateTotalAmount();
        if (amount==0)
           System.out.println("Sorry! Please enter valid movie Id and number of seats");
        else
           System.out.println("Total amount for booking : $" + Math.round(amount*100.0)/100.0);
    }
}
```

****

******

STRINGS   ----->>

EXERCISE ----->>   1

```
class Tester{

    public static String removeWhiteSpaces(String str){
  //Implement your code here and change the return value accordingly
        str = str.replaceAll("\\s", "");
        return str;
 }

 public static void main(String args[]){
  String str = "Hello   How are you   ";
  str = removeWhiteSpaces(str);
  System.out.println(str);
 }
}
```

****
****

ASSIGNMENT -----> 1

```
class Tester{

    public static String moveSpecialCharacters(String str){
  StringBuffer alpha = new StringBuffer(),
        num = new StringBuffer(), special = new StringBuffer();

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

            if(Character.isAlphabetic(str.charAt(i)))
                alpha.append(str.charAt(i));
            else
                special.append(str.charAt(i));
        }
             return (alpha+""+special);
 }

 public static void main(String args[]){
     String str = "He@#$llo!*&";
     System.out.println(moveSpecialCharacters(str));
 }
}
```

****
*****

ASSIGNMENT -----> 2

```
class Tester{
    public static boolean checkPalindrome(String str){
    {
        int i = 0, j = str.length() - 1;
          while (i < j) {
              if (str.charAt(i) != str.charAt(j)) {
                return false;
              }
            i++;
            j--;
        }
        return true;
        }
    }
 public static void main(String args[]){
  String str = "radar";
  if(checkPalindrome(str))
   System.out.println("The string is a palindrome!");
  else
   System.out.println("The string is not a palindrome!");
 }
}
```

****
****

ASSIGNMENT -----> 3

```
class Tester {
    public static String reverseEachWord(String str){
     //Implement your code here and change the return value accordingly
     String words[]=str.split("\\s");
    String reverseWord="";
    for(String w:words){
        StringBuilder sb=new StringBuilder(w);
        sb.reverse();
        reverseWord+=sb.toString()+" ";
    }
    return reverseWord.trim();

 }
 public static void main(String args[]){
     String str = "all cows eat grass";
     System.out.println(reverseEachWord(str));
 }
}
```

****
****

ASSIGNMENT   -----> 4 (STRING)

```
class Tester {

 public static int findHighestOccurrence(String str){
  //Implement your code here and change the return value accordingly

        int len = str.length();
        int count=1;
        for (int i=0; i<len-1; i++) {
            char c=str.charAt(i);
            char a=str.charAt(i+1);
            if(c==a) {
              count=count+1;
               }
        }
        char first=str.charAt(0);
        char last=str.charAt(len-1);


        return count;
 }

 public static void main(String args[]){
     String str = "association";
     System.out.println(findHighestOccurrence(str));
 }
}
```

****
****

```
public class HelloWorld{

     public static void main(String []args){
        System.out.println("Hello World");
```

```
         int[][] evenNum={{1,9,3},{2,3},{3,4,5,6}};
    for(int[] Arr:evenNum)
    {
        for(int ele: Arr) {
            System.out.print(ele+"");
        }
        System.out.println("");
    }
     }
     }
```

```
public class Department {
    private String departmentName;
    private String courses[];
    public String getDepartmentName() {
        return departmentName;
    }
    public void setDepartmentName(String departmentName) {
        this.departmentName = departmentName;
    }
    public String[] getCourses() {
        return courses;
    }
    public void setCourses(String[] courses) {
        this.courses = courses;
    }

    public Department()
    {
        this.departmentName=departmentName;
        this.courses=courses;
    }
    public void displayDeptDetails()
    {
        System.out.println(departmentName);
        for(String arr:courses)
        {
            System.out.println(arr);
        }
    }
}
class Tester {
    public static void main(String[] args)
    {
        Department dept = new Department();
        dept.setDepartmentName("ETA");
        String courses[]={"Java","Python","Data Science","UI"};
        dept.setCourses(courses);
        dept.displayDeptDetails();


    }
```

```
}



class Teacher {
   //Implement your code here
   private String[] teacher;
   private String[] subject;
   private double[] salary;
   public Teacher (String[] teacherNname,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 String setSubject(String[] Subject) {
   this.subject=subject;
   }
   public double getSalary() {
   return salary;
   }
   public void setSalary(double salary) {
   this.salary=salary;
   }
   public void displayDetails() {
      for(int i=0;i<teacherName.length;i++) {
         System.out.println("Name: "+this.teacherName);
          System.out.println("Subjcet: "+this.subject);
           System.out.println("Subjcet: "+this.subject);
      }

   }

}

class Tester {
 public static void main(String[] args) {
    // Implement your code here
    String[] teacherName={"Alex","John","Sam","Maria"};
    String[] subject={"Java","DBMS","Networking","Python"};
    double[] salary={1200.0,800.0,900.0,900.0};
  Teacher obj=new Teacher(teacherName,subject,salary);
    Teacher obj1=new Teacher(teacherName,subject,salary);
    Teacher obj2=new Teacher(teacherName,subject,salary);
 }
```

```
}
```

ARRAY ----->> 2

```
class Tester {

    public static double[] findDetails(double[] salary) {
        double average=0;
         double[] details=new double[3];
        int count1=0;
          int count2=0;
       for(double sal: salary)
        average+=sal;
        average/=(salary.length);
        details[0]=average;
        for(int i=0;i<salary.length;i++){
            if(salary[i]==average)
            {
                  details[0]=average;
            }
        else if(salary[i]>average)
        {
            count1++;
            details[1]=count1;
        }
        else
        {
            count2++;
            details[2]=count2;
        }
        }
        return  details;

    }

    public static void main(String[] args) {
        double[] salary = { 23500.0, 25080.0, 28760.0, 22340.0, 19890.0 };
        double[] details = findDetails(salary);

        System.out.println("Average salary: "+ details[0]);
        System.out.println("Number of salaries greater than the average salary: "+ details[1]);
        System.out.println("Number of salaries lesser than the average salary: "+ details[2]);
    }
}
```

****
****

ARRAY ----->> 3

```
class Tester {

   public static int[] findLeapYears(int year){
       //Implement your code here and change the return value accordingly
       int leapYears[]=new int[15];
       int i=0;
       while(i<15) {
           if((year%400==0)||(year%4==0)&&(year%100!=0))
           {
                   leapYears[i]=year;
                   i=i+1;
           }
           year=year+1;
       }
        return leapYears;
   }

   public static void main(String[] args) {
       int year = 2000;
    int[] leapYears;
    leapYears=findLeapYears(year);
    for ( int index = 0; index<leapYears.length; index++ ) {
        System.out.println(leapYears[index]);
    }
    }
}
```

****

ARRAY ---->> 5

```
public static int[] findNumbers(int num1, int num2) {
  int[] numbers = new int[6];
        int index=0,j=0;
  if(num1<num2) {
      for(int i=num1;i<=num2;i++) {
          int n=i;
          int sum=0;
         while(n!=0) {
          int rem=n%10;

          sum=sum+rem;
          n=n/10;
        }
        if((sum%3==0)&&(i%5==0)) {
          numbers[j]=i;
          j++;
        }
      }
  }

    return numbers;
 }

 public static void main(String[] args) {
  int num1 = -20;
  int num2 = 30;

  int[] numbers = findNumbers(num1, num2);
  if (numbers[0] == 0) {
   System.out.println("There is no such number!");
  } else {
   for (int index = 0; index <= numbers.length - 1; index++) {
    if (numbers[index] == 0) {
     break;
    }
    System.out.println(numbers[index]);
   }
  }

 }
}
```

****

ARRAY ASSIGNMENT ------>> 6

```
class Tester {
 public static int findTotalCount(int[] numbers) {
  //Implement your code here and change the return value accordingly
        int cnt = 0;

        for (int i = 0; i < numbers.length - 1; i++)
        {


            if (numbers[i] == numbers[i + 1])
                cnt++;
        }
        return cnt;

 }

 public static void main(String[] args) {
  int[] numbers = {5,6,6,6,12,1,1,0,0 };
  System.out.println("Count of adjacent occurrence: "+findTotalCount(numbers));
 }
}
```

****
****

STATIC ----->>

ASSIGNMENT ----->> 1

```
class Bill{
    //Implement your code here
    private String paymentMode;
    private String billId;
    private static int counter;
    static {
        counter=9000;
    }
    public String getPaymentMode() {
        return paymentMode;
    }
    public void setPaymentMode(String paymentMode) {
        this.paymentMode=paymentMode;
    }
    public String getBillId() {
        return billId;
    }
    public void setBillId(String billId) {
        this.billId=billId;
    }

    public static int getCounter() {
  return counter;
    }
    public static void setCounter(int counter) {
  Bill.counter = counter;
 }

    public Bill(String paymentMode)
    {
        this.paymentMode=paymentMode;
        this.billId="B"+ ++Bill.counter;

    }
}

class Tester {
    public static void main(String[] args) {
        Bill bill1 = new Bill("DebitCard");
        Bill bill2 = new Bill("PayPal");
        Bill bill3 = new Bill("CreditCard");
        Bill bill4 = new Bill("AmazonPay");
        Bill bill5 = new Bill("GooglePay");

        //Create more objects and add them to the bills array for testing your code

        Bill[] bills = { bill1, bill2, bill3,bill4,bill5 };

        for (Bill bill : bills) {
            System.out.println("Bill Details");
            System.out.println("Bill Id: " + bill.getBillId());
            System.out.println("Payment method: " + bill.getPaymentMode());
            System.out.println();
        }
    }
}
```

****
****

ASSIGNMENT   ----->> 2

```
class Participant {
    //Implement your code here
    private static int counter;
    private String registrationId;
    private String name;
    private long contactNumber;
    private String city;
    static {
        counter=10000;
    }

    public static int getCounter() {
  return counter;
 }
 public static void setCounter(int counter) {
  Participant.counter = counter;
 }

    public String getRegistrationId() {
        return registrationId;
    }
    public void setRegistrationId(String registrationId) {
        this.registrationId=registrationId;
    }
     public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name=name;
    }
    public long getContactNumber() {
        return contactNumber;
    }
    public void setContactNumber(long contactNumber) {
        this.contactNumber=contactNumber;
    }
     public String getCity() {
        return city;
    }
    public void setCity(String city) {
        this.city=city;
    }
    public Participant(String name,long contactNumber,String city) {
        this.registrationId="D" + ++Participant.counter;
        this.name=name;
        this.contactNumber=contactNumber;
        this.city=city;
    }
}

class Tester {

 public static void main(String[] args) {

  Participant participant1 = new Participant("Franklin", 7656784323L, "Texas");
  Participant participant2 = new Participant("Merina", 7890423112L, "New York");

  //Create more objects and add them to the participants array for testing your code

  Participant[] participants = { participant1, participant2 };

  for (Participant participant : participants) {
   System.out.println("Hi "+participant.getName()+"! Your registration id is "+participant.getRegistrationId());
  }

 }
}
```

****
****

ASSIGNMENT ------>> 3

```
class Booking{
    //Implement your code here
    private String customerEmail;
    private int seatsRequired;
    private boolean isBooked;
    private static int seatsAvailable;

    public String getCustomerEmail() {
        return customerEmail;
    }
    public void setCustomerEmail(String customerEmail) {
        this.customerEmail=customerEmail;
    }
    public int getSeatsRequired() {
        return seatsRequired;
    }
    public void setSeatsRequired(int seatsRequired) {
        this.seatsRequired=seatsRequired;
    }
    public static int getSeatsAvailable() {
        return seatsAvailable;
    }
    public static void setSeatsAvailable(int seatsAvailable) {
        Booking.seatsAvailable=seatsAvailable;
    }
    public boolean isBooked() {
        return isBooked;
    }
    public void setIsBooked(boolean isBooked) {
        this.isBooked=isBooked;
    }
    static {
        Booking.seatsAvailable=400;
    }
     public Booking (String customerEmail,int seatsRequired) {
        this.customerEmail=customerEmail;
        this.seatsRequired=seatsRequired;
                  if(Booking.seatsAvailable>=seatsRequired) {
            isBooked= true;
            Booking.seatsAvailable=Booking.seatsAvailable-seatsRequired;
        }
        else {
            isBooked= false;
        }
    }
  }

class Tester {
    public static void main(String[] args) {
        Booking booking1 = new Booking("jack@email.com", 100);
        Booking booking2 = new Booking("jill@email.com", 350);

        //Create more objects and add them to the bookings array for testing your code

        Booking[] bookings = { booking1, booking2 };

        for (Booking booking : bookings) {
            if (booking.isBooked()) {
                System.out.println(booking.getSeatsRequired()+" seats successfully booked for "+booking.getCustomerE
mail());
            }
            else {
                System.out.println("Sorry "+booking.getCustomerEmail()+", required number of seats are not available!")
;
                System.out.println("Seats available: "+Booking.getSeatsAvailable());
            }
        }
    }
}
```

****
*

****

AGGREGATION AND ACCESS MODIFIERS TRY OUT

```
class Subject {
 private String subjectName;
    public String getSubjectName() {
        return this.subjectName;
    }
    public void setSubjectName(String subjectName) {
        this.subjectName=subjectName;
    }
 Subject(String subjectName) {
  this.subjectName = subjectName;
 }
}

class Student {
 private int rollNo;
 private String studentName;
 private Subject subject;
 public int getRollNo() {
     return this.rollNo;
 }
 public void setRollNo(int rolllNo) {
     this.rollNo=rollNo;
 }
 public String getStudentName() {
    return this.studentName=studentName;
 }
 public void setStudentName(String studentName) {
    this.studentName=studentName;
 }


 Student(int rollNo, String studentName, Subject subject) {
  this.rollNo = rollNo;
  this.studentName = studentName;
  this.subject = subject;
 }

 public void displayDetails() {
  System.out.println("Student Name: " + getStudentName());
  System.out.println("Subject Name: " + subject.getSubjectName());

 }

 public static void main(String args[]) {
  Subject subject = new Subject("Maths");
  Student student = new Student(101, "Nate", subject);
  student.displayDetails();
 }
}
```

****

AGGREGATION ----->> 2

```
class Author {
    //Implement your code here
    private String name;
    private String emailId;
    private char gender;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name=name;
    }
    public String getEmailId() {
        return emailId;
    }
    public void setEmailId(String emailId) {
        this.emailId=emailId;
    }
    public char getGender() {
        return gender;
    }
    public void setGender(char gender) {
        this.gender=gender;
    }
    public Author(String name,String emailId,char gender) {
        this.name=name;
        this.emailId=emailId;
        this.gender=gender;
    }
}


class Book {
    //Implement your code here
    private String name;
    private Author author;
    private double price;
    private int quantity;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name=name;
    }
    public Author getAuthor() {
        return author;
    }
    public void setAuthor(Author author) {
        this.author=author;
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double price) {
        this.price=price;
    }
    public int getQuantity() {
        return quantity;
    }
    public void setQuantity(int quantity) {
        this.quantity=quantity;
    }
    public Book(String name,Author author,double price, int quantity) {
    this.name=name;
    this.author=author;
    this.price=price;
    this.quantity=quantity;
    }
    public void displayAuthorDetails() {
        System.out.println("Displaying author details");
        System.out.println("Author name: "+author.getName());
        System.out.println("Author emailId: "+author.getEmailId());
        System.out.println("Author gender: "+author.getGender());
    }
}
```

```
class Tester {
    public static void main(String[] args) {
        //Implement your code here
        Author author1=new Author("Joshua Bloch","joshua@email.com",'M');
     Book book1 =new Book("Effective Java",author1,45d,15);
     book1.displayAuthorDetails();

    }
}
```

```
class Room {
    private int roomNo;
    private int capacity;
    private static int roomCounter;
    static
    {
      roomCounter=500;
    }
    public Room() {
        this.roomNo=this.roomNo+Room.roomCounter++;
        this.capacity=4;
    }
    public int getRoomNo()
    {
        return roomNo;
    }
    public void setRoomNo(int roomNo)
    {
        this.roomNo=roomNo;
    }
     public int getCapacity()
    {
        return capacity;
    }
    public void setCapacity(int capacity)
    {
        this.capacity=capacity;
    }
     public static int getRoomCounter()
    {
        return Room.roomCounter;
    }
    public static void setRoomCounter(int roomCounter)
    {
        Room.roomCounter=roomCounter;
    }


    public String toString(){
        return "Room\nroomNo: "+this.roomNo+"\ncapacity: "+this.capacity;
    }
```

```
}

class Member {
    private int memberId;
    private String name ;
    private Room room;
     public int getMemberId()
    {
        return memberId;
    }
    public void setMemberId(int memberId)
    {
        this.memberId=memberId;
    }
     public String getName()
    {
        return name;
    }
    public void setName(String name)
    {
        this.name=name;
    }
     public Room getRoom()
    {
        return room;
    }
    public void setRoom(Room room)
    {
        this.room=room;
    }
public Member(int memberId,String name)
{
    this.memberId=memberId;
    this.name=name;
    //Room.room=room;
}

//Implement your code here

//Uncomment the below method after implementation before verifying
    //DO NOT MODIFY THE METHOD

    public String toString(){
        return "Member\nmemberId: "+this.memberId+"\nname: "+this.name;
    }

}

class Admin {



        public void assignRoom(Room[] rooms, Member member) {
```

```
        for(Room room: rooms) {
            if(room.getCapacity()!=0) {
                member.setRoom(room);
                room.setCapacity(room.getCapacity()-1);
                break;
            }
        }
    }




//Implement your code here
}


class Tester {
public static void main(String args[]) {
Room room1 = new Room();
Room room2 = new Room();
Room room3 = new Room();
Room room4 = new Room();
Room room5 = new Room();

Room[] totalRooms = { room1, room2, room3, room4, room5 };

Admin admin = new Admin();

Member member1 = new Member(101, "Serena");
Member member2 = new Member(102, "Martha");
Member member3 = new Member(103, "Nia");
Member member4 = new Member(104, "Maria");
Member member5 = new Member(105, "Eva");

Member[] members = { member1, member2, member3, member4, member5 };

for (Member member : members) {
admin.assignRoom(totalRooms, member);
if(member.getRoom()!=null) {
System.out.println("Hi "+member.getName()+"! Your room number is "+member.getRoom().getRoomNo());
}
else {
System.out.println("Hi "+member.getName()+"! No room available");
}
}
}
}
```

****
****

INHERITANCE   ------>>

EXERCISE 1

```
class Camera {
 private String brand;
 private double cost;

 public Camera() {
  this.brand = "Nikon";
 }
    public Camera(String brand,double cost) {
        this.brand=brand;
        this.cost=cost;
    }
 public String getBrand() {
  return brand;
 }
 public void setBrand(String brand) {
  this.brand = brand;
 }
 public double getCost() {
  return cost;
 }
 public void setCost(double cost) {
  this.cost = cost;
 }
}

class DigitalCamera extends Camera {
 private int memory;

 public DigitalCamera(String brand, double cost) {
  super(brand,cost);
  this.memory = 16;
 }

    public int getMemory() {
  return memory;
 }
 public void setMemory(int memory) {
  this.memory = memory;
 }
}

class Tester {
 public static void main(String[] args) {
        DigitalCamera camera = new DigitalCamera("Canon",100);
        System.out.println(camera.getBrand()+" "+camera.getCost()+" "+camera.getMemory());
    }
}
```

****

****

INHERITANCE

ASSIGNMENT ----->> 1

```
class Employee {

   //Implement your code here
   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;
   }
   //Uncomment the below method after implementation before verifying
   //DO NOT MODIFY THE METHOD

   public String toString(){
      return "Employee\nemployeeId: "+this.getEmployeeId()+"\nemployeeName: "+this.getEmployeeName()+"\nsalary: "+this.getSalary();
   }

}


class PermanentEmployee extends Employee {
```

```
//Implement your code here
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() {
    int variableComponent=0;
    if(experience<3) {
        variableComponent=0;
    }

    else if((experience>=3)&&(experience<5)) {
        variableComponent=5;
    }

     else if((experience>=5)&&(experience<10)) {
        variableComponent=7;
    }

     else if(experience>=10) {
        variableComponent=12;
    }
    this.setSalary((float)(basicPay+hra+(basicPay*variableComponent*0.01)));
}

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;
}
//Uncomment the below method after implementation before verifying
```

```
    //DO NOT MODIFY THE METHOD

    public String toString(){
        return "PermanentEmployee\nemployeeId: "+this.getEmployeeId()+"\nemployeeName: "+this.getEmployeeName()+"\nsalary: "+this.getSalary()+"\nbasicPay: "+this.getBasicPay()+"\nhra: "+this.getHra()+"\nexperience: "+this.getExperience();
    }

}

class ContractEmployee extends Employee {

    //Implement your code here
    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 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 void calculateSalary() {
        this.setSalary(hoursWorked*(float)wage);
    }
    //Uncomment the below method after implementation before verifying
    //DO NOT MODIFY THE METHOD

    public String toString(){
        return "ContractEmployee\nemployeeId: "+this.getEmployeeId()+"\nemployeeName: "+this.getEmployeeName()+"\nsalary: "+this.getSalary()+"\nwage: "+this.getWage()+"\nhoursWorked: "+this.getHoursWorked();
    }

}

class Tester {

    public static void main(String[] args) {
        PermanentEmployee permanentEmployee = new PermanentEmployee(711211, "Rafael", 1850, 115, 3.5f);
        permanentEmployee.calculateMonthlySalary();
        System.out.println("Hi "+permanentEmployee.getEmployeeName()+", your salary is $"+Math.round(permanentEmployee.getSalary()*100)/100.0);

        ContractEmployee contractEmployee = new ContractEmployee(102, "Jennifer", 16, 90);
        contractEmployee.calculateSalary();
        System.out.println("Hi "+contractEmployee.getEmployeeName()+", your salary is $"+Math.round(contractEmployee.getSalary()*100)/100.0);

        //Create more objects for testing your code
    }

}
```

****
****

METHOD OVERLOADING EXERCISE ----->> 1

```
class Point {
    //Implement your code here
    private double xCoordinate;
    private double yCoordinate;

    public Point(double xCoordinate,double yCoordinate) {
        this.xCoordinate=xCoordinate;
        this.yCoordinate=yCoordinate;
    }

    public double calculateDistance() {
        double x1=0.0,y1=0.0;
        double x2=this.xCoordinate,y2=this.yCoordinate;
        double distance=Math.sqrt((x2-x1)*(x2-x1) + (y2-y1)*(y2-y1));
        return Math.round(distance*100.0)/100.0;
    }
    public double calculateDistance(Point point) {
        double x1=this.xCoordinate,y1=this.yCoordinate;
        double x2=point.xCoordinate,y2=point.yCoordinate;
        double distance=Math.sqrt((x2-x1)*(x2-x1) + (y2-y1)*(y2-y1));
        return Math.round(distance*100.0)/100.0;
    }
    public double getxCoordinate() {
        return xCoordinate;
    }
    public void setxCoordinate(double xCoordinate) {
        this.xCoordinate=xCoordinate;
    }
    public double getyCoordinate() {
        return yCoordinate;
    }
    public void setyCoordinate(double yCoordinate) {
        this.yCoordinate=yCoordinate;
    }
}

class Tester {

 public static void main(String[] args) {
  Point point1 = new Point(3.5, 1.5);
        Point point2 = new Point(6, 4);

        System.out.println("Distance of point1 from origin is "+point1.calculateDistance());
        System.out.println("Distance of point2 from origin is "+point2.calculateDistance());
        System.out.println("Distance of point1 from point2 is "+point1.calculateDistance(point2));

        //Create more objects for testing your code

 }
}
```

****
****

METHOD OVERLOADING   ------>>> 1

```
class Bill{
    //Implement your code here
    double price=0;
    public double findPrice(int itemId) {

    if (itemId==1001) {
        price = 25;
    }
    else if(itemId==1002) {
       price=20;
    }
    else if(itemId==1003) {
        price=23;
    }
    else if(itemId==1004) {
       price=18;
    }
    else {
       price =0;
    }
    return price;
    }

    public double findPrice(String brandName, String itemType, int size) {
        if(brandName=="Puma") {
         if(itemType=="T-shirt"&&(size==34||size==36)) {
            price=25;
         }
         else if(itemType=="Skirt"&&(size==38||size==40)) {
            price=20;
         }
         else {
            price=0;
         }
      }
      else if(brandName=="Reebok") {
         if(itemType=="T-shirt"&&(size==34||size==36)) {
            price=23;
         }
         else if(itemType=="Skirt"&&(size==38||size==40)) {
            price=18;
         }
         else {
            price=0;
         }
      }
      return price;
   }
}

class Tester {

 public static void main(String[] args) {

  Bill bill = new Bill();

  double price = bill.findPrice(1001);
  if(price>0)
      System.out.println("Price of the selected item is $"+price);
  else
      System.out.println("The Item Id is invalid");

  price = bill.findPrice("Reebok","Skirt",38);
  if(price>0)
      System.out.println("Price of the selected item is $"+price);
  else
      System.out.println("The values are not valid");
 }

}
```

```
********************************************************************************************
*******************************
```

```
class Bill{
   //Implement your code here
   public double findPrice(int itemId) {
       int price=0;
   if (itemId==1001) {
       price = 25;
   }
   else if(itemId==1002) {
      price=20;
   }
   else if(itemId==1003) {
       price=23;
   }
   else if(itemId==1004) {
      price=18;
   }
   else {
      price =0;
   }
   return price;
   }

   public double findPrice(String brandName, String itemType, int size) {
       double price=0;
       if((brandName=="Puma")&&(itemType=="T-shirt")&&((size==34)||(size==36)))
       {
           price=25;
       }
       else if((brandName=="Puma")&&(itemType=="Skirt")&&((size==38)||(size==40)))
       {
           price=20;
       }
        else if((brandName=="Reebok")&&(itemType=="T-shirt")&&((size==34)||(size==36)))
       {
           price=23;
       }
        else if((brandName=="Reebok")&&(itemType=="Skirt")&&((size==38)||(size==40)))
       {
           price=18;
       }
       else {
           price=0;
       }
       return price;
   }
}

class Tester {

 public static void main(String[] args) {

  Bill bill = new Bill();

  double price = bill.findPrice(1001);
  if(price>0)
      System.out.println("Price of the selected item is $"+price);
  else
```

```
      System.out.println("The Item Id is invalid");

  price = bill.findPrice("Reebok","Skirt",38);
  if(price>0)
      System.out.println("Price of the selected item is $"+price);
  else
      System.out.println("The values are not valid");
 }

}
```

****
****

FINAL   ----->>   EXERCISE ---->> 1

```
class Student{
    //Implement your code here
    private final int STIPEND = 100;
    private int studentId;
    private int aggregateMarks;

    public int getStudentId() {
        return studentId;
    }

    public void setStudentId(int studentId) {
        this.studentId = studentId;
    }

    public int getAggregateMarks() {
        return aggregateMarks;
    }

    public void setAggregateMarks(int aggregateMarks) {
        this.aggregateMarks = aggregateMarks;
    }

    public int getSTIPEND() {
        return STIPEND;
    }

    public double calculateTotalStipend() {
        double totalStipend=0;
        if((aggregateMarks>=85)&&(aggregateMarks<90)) {
            totalStipend=10;
        }
        else if((aggregateMarks>=90)&&(aggregateMarks<95)) {
           totalStipend =15;
        }
        else if((aggregateMarks>=95)&&(aggregateMarks<=100)) {
            totalStipend=20;
        }
        totalStipend=STIPEND+totalStipend;
        return totalStipend;
    }

}

class Tester {

 public static void main(String[] args) {
  Student student1 = new Student();
  student1.setStudentId(1212);
  student1.setAggregateMarks(93);

  double totalStipend = student1.calculateTotalStipend();
  System.out.println("The final stipend of " + student1.getStudentId()+" is $" + totalStipend);

  Student student2 = new Student();
  student2.setStudentId(1222);
  student2.setAggregateMarks(84);

  totalStipend = student2.calculateTotalStipend();
  System.out.println("The final stipend of " + student2.getStudentId()+" is $" + totalStipend);
 }

}
```

****
****

METHOD OVER RIDDING

EXERISE ----->> 1

```
class User{
  private  int id;
  private  String userName;
   private String emailId;
  private  double walletBalance;
public User(int id,String userName,String emailId, double walletBalance)
{
    this.id=id;
    this.userName=userName;
     this.emailId=emailId;
     this.walletBalance=walletBalance;
}
public int getId()
{
    return id;
}
public void setId(int id)
{
    this.id=id;
```

```
}
public String getUserName()
{
   return userName;
}
public void setUserName(String userName)
{
  this.userName=userName;
}
public String getEmailId()
{
   return emailId;
}
public void setEmailId(String emailId)
{
   this.emailId=emailId;
}
public double getWalletBalance()
{
   return walletBalance;
}
public void setWalletBalance( double walletBalance)
{
   this.walletBalance=walletBalance;
}
public boolean makePayment(double billAmount)
{
  // this.billAmount=billAmount;
   if(billAmount<=walletBalance)
    {
      this.setWalletBalance(walletBalance-billAmount);
       return true;
    }
   else
    {
       return false;
    }
}
}

class PremiumUser extends User{
   private int rewardPoints;
  public PremiumUser(int id,String userName,String emailId, double walletBalance)
{
   super(id,userName,emailId,walletBalance);
  this.rewardPoints=rewardPoints;
}
public int getRewardPoints()
{
   return rewardPoints;
}
public void setRewardPoints(int rewardPoints)
{
   this.rewardPoints+=rewardPoints;
```

```
}
public boolean makePayment(double billAmount)
{
    boolean output=super.makePayment(billAmount);
    if(output==true)
    {

        this.setRewardPoints((int)(0.1*billAmount));
        return true;
    }
    else
    return false;
    // this.billAmount=billAmount;
   /* if(billAmount<=getWalletBalance)
    {
        this.setWalletBalance(getWalletBalance - billAmount);
        this.setRewardPoints((int)(0.1*billAmount));
        return true;
    }
    else
    {
        return false;
    }
    */
}
}

class Tester {

    public static void main(String[] args) {

        User user = new User(101, "Joe", "joe@abc.com", 100);

PremiumUser premiumUser = new PremiumUser(201, "Jill", "jill@abc.com", 300);

processPayment(user, 70);

processPayment(premiumUser, 150);

processPayment(premiumUser, 80);

processPayment(premiumUser, 120);

    }

    public static void processPayment(User user, double billAmount) {
        if (user.makePayment(billAmount)) {
System.out.println("Congratulations " + user.getUserName() + ", payment of $" + billAmount + " was successful!");

} else {
System.out.println("Sorry " + user.getUserName() + ", you do not have enough balance to pay the bill!");
}
System.out.println("Your wallet balance is $" + user.getWalletBalance());
```

```
if (user instanceof PremiumUser) {
PremiumUser premiumUser = (PremiumUser) user;
System.out.println("You have " + premiumUser.getRewardPoints() + " points!");
}
System.out.println();
    }
}
```

****
****

METHOD OVER RIDING ---->>

ASSIGNMENT ---->> 1

```
class Faculty{
    private String name;
 private float basicSalary;
 private float bonusPercentage;
 private float carAllowancePercentage;
 public Faculty(String name,float basicSalary){
  this.name= name;
  this.basicSalary= basicSalary;
  this.bonusPercentage=4f;
  this.carAllowancePercentage= 2.5f;
 }

 public String getName() {
  return name;
 }
 public void setName(String name) {
  this.name = name;
 }
 public float getBasicSalary() {
  return basicSalary;
 }
 public void setBasicSalary(float basicSalary) {
  this.basicSalary = basicSalary;
 }
 public float getBonusPercentage() {
  return bonusPercentage;
 }
 public void setBonusPercentage(float bonusPercentage) {
  this.bonusPercentage = bonusPercentage;
 }
 public float getCarAllowancePercentage() {
  return carAllowancePercentage;
 }
 public void setCarAllowancePercentage(float carAllowancePercentage) {
  this.carAllowancePercentage = carAllowancePercentage;
 }
 public double calculateSalary(){
  return this.getBasicSalary()*(1+(this.getBonusPercentage()/100)+(this.getCarAllowancePercentage()/100));
 }

}

class OfficeStaff extends Faculty{
    private String designation;
 public OfficeStaff(String name, float basicSalary, String designation){
  super(name,basicSalary);
  this.designation= designation;
 }


 public String getDesignation() {
  return designation;
 }
 public void setDesignation(String designation) {
  this.designation = designation;
 }
 public double calculateSalary(){
  double basic =super.calculateSalary();
  double salary=0;
  if(this.getDesignation().equals("Accountant")){
   salary= basic+10000.0;
  }
  else if(this.getDesignation().equals("Clerk")){
   salary= basic+7000.0;
  }
  else if(this.getDesignation().equals("Peon")){
   salary= basic+4500.0;
  }
  else{
   salary= basic;
  }

  return salary;
  }
}

class Teacher extends Faculty{
    //Implement your code here
 private String qualification;
 public Teacher(String name, float basicSalary, String qualification){
  super(name,basicSalary);
  this.qualification= qualification;
 }


 public String getQualification() {
  return qualification;
 }
 public void setQualification(String qualification) {
  this.qualification = qualification;
 }
  public double calculateSalary(){
  double basic =super.calculateSalary();
  double salary=0;
  if(this.getQualification().equals("Doctoral")){
   salary= basic+20000.0;
  }
  else if(this.getQualification().equals("Masters")){
   salary= basic+18000.0;
  }
  else if(this.getQualification().equals("Bachelors")){
   salary= basic+15500.0;
  }
  else if(this.getQualification().equals("Associate")){
   salary= basic+10000.0;
  }
  else{
   salary= basic;
  }

  return salary;
 }

}


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

  Teacher teacher = new Teacher("Caroline", 30500f, "Masters");
  OfficeStaff officeStaff = new OfficeStaff("James", 24000f, "Accountant");

  System.out.println("Teacher Details\n***************");
  System.out.println("Name: "+teacher.getName());
  System.out.println("Qualification: "+teacher.getQualification());
  System.out.println("Total salary: $" + Math.round(teacher.calculateSalary()*100)/100.0);
  System.out.println();

  System.out.println("Office Staff Details\n***************");
  System.out.println("Name: "+officeStaff.getName());
  System.out.println("Designation: "+officeStaff.getDesignation());
  System.out.println("Total salary: $" + Math.round(officeStaff.calculateSalary()*100)/100.0);

        //Create more objects for testing your code

 }
}
```

****
****

METHOD OVERRIDING

ASSIGNMENT ----->> 2

```
class Event{
 //Implement your code here
 private String eventName;
 private String participantName;
 private double registrationFee;
 public Event(String eventName, String participantName){
  this.eventName= eventName;
  this.participantName= participantName;
 }
 public void registerEvent(){
  if(this.getEventName().equals("Singing")){
   this.setRegistrationFee(8);
  }
  else if(this.getEventName().equals("Dancing")){
   this.setRegistrationFee(10);
  }
  else if(this.getEventName().equals("DigitalArt")){
   this.setRegistrationFee(12);
  }
  else if(this.getEventName().equals("Acting")){
   this.setRegistrationFee(15);
  }
  else{
   this.setRegistrationFee(0);
  }
 }
 public String getEventName() {
  return eventName;
 }
 public void setEventName(String eventName) {
  this.eventName = eventName;
 }
 public String getParticipantName() {
  return participantName;
 }
 public void setParticipantName(String participantName) {
  this.participantName = participantName;
 }
 public double getRegistrationFee() {
  return registrationFee;
 }
 public void setRegistrationFee(double registrationFee) {
  this.registrationFee = registrationFee;
 }

}

class SoloEvent extends Event{
    //Implement your code here
 private int participantNo;
 public SoloEvent(String eventName, String participantName, int participantNo){
  super(eventName, participantName);
  this.participantNo= participantNo;
 }

    public void registerEvent(){
  super.registerEvent();
 }
    public int getParticipantNo() {
 return participantNo;
    }
    public void setParticipantNo(int participantNo) {
 this.participantNo = participantNo;
    }
}

class TeamEvent extends Event{
    //Implement your code here
 private int noOfParticipants;
 private int teamNo;
 public TeamEvent(String eventName, String participantName, int noOfParticipants, int teamNo){
  super(eventName, participantName);
  this.noOfParticipants= noOfParticipants;
  this.teamNo= teamNo;
 }

public void registerEvent(){
  double fee=0;
  if(this.getNoOfParticipants()>1){
  if(this.getEventName().equals("Singing")){
   fee= (double)(this.getNoOfParticipants()*4);
   this.setRegistrationFee(fee);
  }
  else if(this.getEventName().equals("Dancing")){
   fee= (double)(this.getNoOfParticipants()*6);
   this.setRegistrationFee(fee);
  }
  else if(this.getEventName().equals("DigitalArt")){
   fee= (double)(this.getNoOfParticipants()*8);
   this.setRegistrationFee(fee);
  }
  else if(this.getEventName().equals("Acting")){
   fee= (double)(this.getNoOfParticipants()*10);
   this.setRegistrationFee(fee);
  }
  else{
   this.setRegistrationFee(0);
  }
  }

 }
public int getNoOfParticipants() {
 return noOfParticipants;
}
public void setNoOfParticipants(int noOfParticipants) {
 this.noOfParticipants = noOfParticipants;
```

```
}
public int getTeamNo() {
 return teamNo;
}
public void setTeamNo(int teamNo) {
 this.teamNo = teamNo;
}

}

class Tester {

      public static void main(String[] args) {

        SoloEvent soloEvent = new SoloEvent("Dancing", "Jacob", 1);
  soloEvent.registerEvent();
  if (soloEvent.getRegistrationFee() != 0) {
   System.out.println("Thank You " + soloEvent.getParticipantName()
     + " for your participation! Your registration fee is $" + soloEvent.getRegistrationFee());
   System.out.println("Your participant number is " + soloEvent.getParticipantNo());

  } else {
   System.out.println("Please enter a valid event");
  }

  System.out.println();
  TeamEvent teamEvent = new TeamEvent("Acting", "Serena", 5, 1);
  teamEvent.registerEvent();
  if (teamEvent.getRegistrationFee() != 0) {
   System.out.println("Thank You " + teamEvent.getParticipantName()
     + " for your participation! Your registration fee is $" + teamEvent.getRegistrationFee());
   System.out.println("Your team number is " + teamEvent.getTeamNo());
  } else {
   System.out.println("Please enter a valid event");
  }
 }
}
```

****
****

FINAL

ASSIGNMENT   ----->> 1

```
class Circle{
 private final double PI=3.14;
 private double diameter;
 private double circumference;
 private double area;
 public Circle(double diameter) {
  this.diameter=diameter;
 }

 public void calculateCircumference() {
  circumference=this.getDiameter()*(this.PI);
  this.setCircumference(circumference);
 }

 public void calculateArea() {
  double dia=this.getDiameter()/2;
  area=(this.PI)*(dia*dia);
  this.setArea(area);
 }

 public double getDiameter() {
  return diameter;
 }

 public void setDiameter(double diameter) {
  this.diameter = diameter;
 }

 public double getCircumference() {
  return circumference;
 }

 public void setCircumference(double circumference) {
  this.circumference = circumference;
 }

 public double getArea() {
  return area;
 }

 public void setArea(double area) {
  this.area = area;
 }

 public double getPI() {
  return PI;
 }


}
 class FinalAssignment {

 public static void main(String[] args) {
  // TODO Auto-generated method stub
  Circle circle1 = new Circle(12.2);
        Circle circle2 = new Circle(33.2);

        //Create more objects of Circle class and add to the array given below for testing your code
        Circle[] circles = {circle1, circle2};
        for (Circle circle : circles) {

            circle.calculateCircumference();
            circle.calculateArea();

            System.out.println("Diameter of the circle is "+circle.getDiameter());
            System.out.println("Circumference of the circle is " + Math.round(circle.getCircumference()*100)/100.0);
            System.out.println("Area of the circle is " + Math.round(circle.getArea()*100)/100.0);
            System.out.println();
        }
 }

}
```

&&&&&&&&&&&&&&&&&&&&&&&&&

```
class Circle{

    private final double PI=3.14;
    private double diameter;
    private double circumference;
    private double area;
    public double getDiameter() {
        return diameter;
    }
    public void setDiameter(double diameter) {
        this.diameter = diameter;
    }
    public double getCircumference() {
        return circumference;
    }
    public void setCircumference(double circumference) {
        this.circumference = circumference;
    }
    public double getArea() {
        return area;
    }
    public void setArea(double area) {
        this.area = area;
    }
    public double getPI() {
        return PI;
    }
   public Circle(double diameter) {
        this.diameter=diameter;
    }
    public void calculateCircumference() {
      this.circumference=this.PI*this.getDiameter();
    }
    public void calculateArea() {
     double r=this.getDiameter()/2;
     this.area=this.PI*r*r;
    }
}
 class FinalAssignment {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Circle circle1 = new Circle(10.2);
        Circle circle2 = new Circle(5.7);
        //Create more objects of Circle class and add to the array given below for testing your code
        Circle[] circles = {circle1, circle2};
        for (Circle circle : circles) {
            circle.calculateCircumference();
            circle.calculateArea();
            System.out.println("Diameter of the circle is "+circle.getDiameter());
            System.out.println("Circumference of the circle is " + Math.round(circle.getCircumference()*100)/100.0);
            System.out.println("Area of the circle is " + Math.round(circle.getArea()*100)/100.0);
            System.out.println();
        }
    }
}
```

****
****

ABSTRACT ---->>

EXERCISE ----->> 1

```
abstract class Student{
    //Implement your code here
    private String studentName;
    private int[] testScores;
    private String testResult;
    public Student(String studentName) {
        this.studentName=studentName;
        testScores=new int[4];
    }
    abstract public void generateResult();
    public void setTestScore(int testNumber, int testScore) {
        this.testScores[testNumber]=testScore;
    }
    public String getStudentName() {
        return studentName;
    }
    public void setStudentName(String studentName) {
        this.studentName=studentName;
    }
    public int[] getTestScores() {
        return testScores;
    }
    public String getTestResult() {
        return testResult;
    }
    public void setTestResult(String testResult) {
        this.testResult=testResult;
    }
}

class UndergraduateStudent extends Student{
    //Implment your code here
    public UndergraduateStudent(String studentName) {
        super(studentName);
    }
    public void generateResult() {
       int[] testScores=getTestScores();
    int scores = 0;
    int a=testScores.length;
    for(int i=0;i<a;i++) {
    scores=scores+testScores[i];
    }
    double avg=(double)scores/a;
    if(avg>=60) {
    setTestResult("Pass");
    }
    else if(avg<60) {
    setTestResult("Fail");
    }
 }
}
class GraduateStudent extends Student{
    //Implment your code here
    public GraduateStudent(String studentName) {
        super(studentName);
    }
    public void generateResult() {
       int[] testScores=getTestScores();
    int scores = 0;
    int a=testScores.length;
    for(int i=0;i<a;i++) {
    scores=scores+testScores[i];
    }
    double avg = (double) scores/a;
    if(avg>=70) {
    setTestResult("Pass");
    }
    else if(avg<70) {
    setTestResult("Fail");
    }
    }
}

class Tester {

    public static void main(String[] args) {
        UndergraduateStudent undergraduateStudent = new UndergraduateStudent("Philip");
        undergraduateStudent.setTestScore(0, 70);
        undergraduateStudent.setTestScore(1, 69);
        undergraduateStudent.setTestScore(2, 71);
        undergraduateStudent.setTestScore(3, 55);

        undergraduateStudent.generateResult();

        System.out.println("Student name: "+undergraduateStudent.getStudentName());
        System.out.println("Result: "+undergraduateStudent.getTestResult());

        System.out.println();

        GraduateStudent graduateStudent = new GraduateStudent("Jerry");
        graduateStudent.setTestScore(0, 70);
        graduateStudent.setTestScore(1, 69);
        graduateStudent.setTestScore(2, 71);
        graduateStudent.setTestScore(3, 55);

        graduateStudent.generateResult();

        System.out.println("Student name: "+graduateStudent.getStudentName());
        System.out.println("Result : "+graduateStudent.getTestResult());

        //Create more objects of the classes for testing your code
    }
}
```

****
****

ASSIGNMENT ---->>1

```
abstract class Payment{
 private int customerId;
 protected String paymentId;
 protected double serviceTaxPercentage;
 public int getCustomerId() {
  return customerId;
 }
 public void setCustomerId(int customerId) {
  this.customerId = customerId;
 }
 public String getPaymentId() {
  return paymentId;
 }
 public void setPaymentId(String paymentId) {
  this.paymentId = paymentId;
 }
 public double getServiceTaxPercentage() {
  return serviceTaxPercentage;
 }
 public void setServiceTaxPercentage(double serviceTaxPercentage) {
  this.serviceTaxPercentage = serviceTaxPercentage;
 }
 public Payment(int customerId) {
  this.customerId=customerId;
 }
 public abstract double payBill(double amount);
}

class DebitCardPayment extends Payment{
 private static int counter=1000;
 private double discountPercentage;

 public DebitCardPayment(int customerId) {
  super(customerId);
  this.setPaymentId("D"+ ++counter);
 }

 public static int getCounter() {
  return counter;
 }

 public static void setCounter(int counter) {
  DebitCardPayment.counter = counter;
 }

 public double getDiscountPercentage() {
  return discountPercentage;
 }

 public void setDiscountPercentage(double discountPercentage) {
  this.discountPercentage = discountPercentage;
 }

 public double payBill(double amount) {
  double tax,discount,billAmount=0.0;
  double serviceTaxPercentage=0;
  if(amount<=500) {
   serviceTaxPercentage=2.5;
   discountPercentage=1;
  }
  else if(amount>500 && amount<=1000 ) {
   serviceTaxPercentage=4;
   discountPercentage=2;
  }
  else if(amount>1000 ) {
   serviceTaxPercentage=5;
   discountPercentage=3;
  }
      this.setServiceTaxPercentage(serviceTaxPercentage);
      discount=(amount*discountPercentage)/100;
   tax=amount+(amount*serviceTaxPercentage/100);
   billAmount=tax-discount;
      return billAmount;
 }
```

```
}

class CreditCardPayment extends Payment{
 private static int counter=1000;

 public CreditCardPayment(int customerId) {
  super(customerId);
  this.setPaymentId("C"+ ++counter);
 }
 public static int getCounter() {
  return counter;
 }
 public static void setCounter(int counter) {
  CreditCardPayment.counter = counter;
 }

 public double payBill(double amount) {
  double billAmount=0.0;
  double serviceTaxPercentage=0;
  if(amount<=500) {
      serviceTaxPercentage=3;
  }
  else if(amount>500 && amount<=1000 ) {
   serviceTaxPercentage=5;
  }
  else if(amount>1000 ) {
   serviceTaxPercentage=6;
  }
  this.setServiceTaxPercentage(serviceTaxPercentage);
  billAmount=amount+(amount*(serviceTaxPercentage/100));
  return billAmount;
 }
}

 class AbstractAssignment {

 public static void main(String[] args) {
  // TODO Auto-generated method stub
  DebitCardPayment debitCardPayment = new DebitCardPayment(101);
        double billAmount=Math.round(debitCardPayment.payBill(500)*100)/100.0;
  System.out.println("Customer Id: " + debitCardPayment.getCustomerId());
  System.out.println("Payment Id: " + debitCardPayment.getPaymentId());
  System.out.println("Service tax percentage: " + debitCardPayment.getServiceTaxPercentage());
  System.out.println("Discount percentage: " + debitCardPayment.getDiscountPercentage());
  System.out.println("Total bill amount: " + billAmount);

  CreditCardPayment creditCardPayment = new CreditCardPayment(102);
        billAmount=Math.round(creditCardPayment.payBill(1000)*100)/100.0;
  System.out.println("Customer Id: " + creditCardPayment.getCustomerId());
  System.out.println("Payment Id: " + creditCardPayment.getPaymentId());
  System.out.println("Service tax percentage: " + creditCardPayment.getServiceTaxPercentage());
  System.out.println("Total bill amount: " + billAmount);
 }
```

}

****
****

INTERFACE

EXERCISE ---->> 1

```
interface Tax{
   //Implement your code here
   double calculateTaxPrice(double price);
}

class PurchaseDetails{
   //Implement your code here
   private String purchaseId;
   private String paymentType;
   private double taxPercentage;
   public PurchaseDetails(String purchaseId, String paymentType) {
      this.purchaseId=purchaseId;
      this.paymentType=paymentType;
   }

   public double calculateTax(double price) {
      //super(price);
      //this.price=price;
      double tax=0;
      if (paymentType=="Debit Card") {
         tax=2;
         //this.setTaxPercentage(tax);
      }
      else if(paymentType=="Credit Card") {
         tax=3;
         //this.setTaxPercentage(tax);
      }
      else {
         tax=4;
         //this.setTaxPercentage(tax);
      }
      //tax=price*(tax);
      price=price+(price*(tax/100));
      this.setTaxPercentage(tax);
      return price;
   }

   public String getPurchaseId() {
      return purchaseId;
   }
   public void setPurchaseId(String purchaseId) {
      this.purchaseId=purchaseId;
```

```
    }
    public String getPaymentType() {
        return paymentType;
    }
    public void setPaymentType(String paymentType) {
        this.paymentType=paymentType;
    }
    public double getTaxPercentage() {
        return taxPercentage;
    }
    public void setTaxPercentage(double taxPercentage) {
        this.taxPercentage=taxPercentage;
    }


}

class Seller{
    //Implement your code here
    private String location;
    private double taxPercentage;
    public String getLocation() {
        return location;
    }
    public void setLocation(String location) {
        this.location=location;
    }
    public double getTaxPercentage() {
        return taxPercentage;
    }
    public void setTaxPercentage(double taxPercentage) {
        this.taxPercentage=taxPercentage;
    }
    public double calculateTax(double price) {
        //this.price=price;
        //super(price);
        double tax=0;
        if(location=="Middle east") {
            tax=15;
            //this.setTaxPercentage(15);
        }
        else if(location=="Europe") {
            tax=25;
            //this.setTaxPercentage(25);
        }
        else if(location=="Canada") {
            tax=22;
            //this.setTaxPercentage(22);
        }
        else if(location=="Japan") {
            tax=12;
            //this.setTaxPercentage(12);
        }
        //tax=price*tax;
        //price=price+tax;
        //return tax;
        this.setTaxPercentage(tax);
        return (price*(tax/100));

    }
    public Seller(String location) {
        this.location=location;
    }
}

class Tester{
    public static void main(String args[]) {
  System.out.println("Purchase Details\n***************");
  PurchaseDetails purchaseDetails = new PurchaseDetails("P1001","Credit Card");
  System.out.println("Total purchase amount: " + Math.round(purchaseDetails.calculateTax(100)*100)/100.0);
  System.out.println("Tax percentage: "+purchaseDetails.getTaxPercentage());

        System.out.println("Seller Details\n***************");
  Seller seller = new Seller("Canada");
  System.out.println("Tax to be paid by the seller: " + Math.round(seller.calculateTax(100)*100)/100.0);
  System.out.println("Tax percentage: "+seller.getTaxPercentage());

  //Create more objects for testing your code
 }
}
```

****
****

ASSIGNMENT ---->> 1

```
interface Testable{
    //Implement your code here
    boolean testCompatibility();
}

class Mobile {
    //Implement your code here
    private String name;
    private String brand;
    private String operatingSystemName;
    private String operatingSystemVersion;
    public Mobile(String name, String brand, String operatingSystemName, String operatingSystemVersion) {
        this.name=name;
        this.brand=brand;
        this.operatingSystemName=operatingSystemName;
        this.operatingSystemVersion=operatingSystemVersion;
    }
    public String getName() {
        return name;
    }
```

```
   public void setName(String name) {
      this.name=name;
   }
   public String getBrand() {
      return brand;
   }
   public void setBrand(String brand) {
      this.brand=brand;
   }
   public String getOperatingSystemName() {
      return operatingSystemName;
   }
   public void setOperatingSystemName(String operatingSystemName) {
      this.operatingSystemName=operatingSystemName;
   }
   public String getOperatingSystemVersion() {
      return operatingSystemVersion;
   }
   public void setOperatingSystemVersion(String operatingSystemVersion) {
      this.operatingSystemVersion=operatingSystemVersion;
   }
}

class SmartPhone extends Mobile implements Testable {
   //Implement your code here
   private String networkGeneration;
   public SmartPhone(String name, String brand, String operatingSystemName, String operatingSystemVersion, String networkGeneration) {
      super(name,brand,operatingSystemName,operatingSystemVersion);
      this.networkGeneration=networkGeneration;
   }
    public String getNetworkGeneration() {
      return networkGeneration;
   }
   public void setNetworkGeneration(String networkGeneration) {
      this.networkGeneration=networkGeneration;
   }
   public boolean testCompatibility() {
      //if(operatingSystemName=="Saturn") {
      if((getOperatingSystemName()=="Saturn")&&(networkGeneration=="3G")&&((getOperatingSystemVersion()=="1.1")||(getOperatingSystemVersion()=="1.2")||(getOperatingSystemVersion()=="1.3")))
      {
         return true;
      }
      else if((getOperatingSystemName()=="Saturn")&&(networkGeneration=="4G")&&((getOperatingSystemVersion()=="1.3")||(getOperatingSystemVersion()=="1.2")))
      {
         return true;
      }
      else if((getOperatingSystemName()=="Saturn")&&(networkGeneration=="5G")&&(getOperatingSystemVersion()=="1.3"))
      {
         return true;
      }
```

```
       else if((getOperatingSystemName()=="Gara")&&(networkGeneration=="3G")&&((getOperatingSystemVersion()=="EXRT.1")||(getOperatingSystemVersion()=="EXRT.2")||(getOperatingSystemVersion()=="EXRU.1")))
       {
          return true;
       }
       else if((getOperatingSystemName()=="Gara")&&(networkGeneration=="4G")&&((getOperatingSystemVersion()=="EXRT.2")||(getOperatingSystemVersion()=="EXRU.1")))
       {
          return true;
       }
       else if((getOperatingSystemName()=="Gara")&&(networkGeneration=="5G")&&(getOperatingSystemVersion()=="EXRU.1"))
       {
          return true;
       }
       else
       return false;
    }

}
```

****
****

EXCEPTION EXERCISE ---->> 1

```
class InvalidAgeException extends Exception{
 public InvalidAgeException(String message){
  super(message);

 }
}
 class InvalidNameException extends Exception{
 public InvalidNameException(String message){
  super(message);

 }
}
 class InvalidJobProfileException extends Exception{
   public InvalidJobProfileException(String message){
       super(message);

 }
}



class Applicant{

 private String name;
 private String jobProfile;
 private int age;
 public String getName() {
    return name;
 }

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

 public String getJobProfile() {
     return jobProfile;
 }

 public void setJobProfile(String jobProfile) {
    this.jobProfile = jobProfile;
 }

 public int getAge() {
    return age;
 }

 public void setAge(int age) {
    this.age = age;
 }
}


class Validator  {
 public boolean  validateName(String name) {
  if ( name==null || name.isEmpty() || name==" ")
   return false;
  else
   return true;
 }

 public boolean validateJobProfile(String jobProfile) {
if (jobProfile.equalsIgnoreCase("Associate") || jobProfile.equalsIgnoreCase("Clerk") || jobProfile.equalsIgnoreCase(
"Executive") ||jobProfile.equalsIgnoreCase("Officer") ) {
 return true;
}else
 return false;
}
public boolean validateAge(int age) {
 if(age>=18 && age<=30) {
  return true;
 }
 else
  return false;
}
 public void validate(Applicant applicant) throws InvalidNameException,InvalidJobProfileException,InvalidAgeExc
eption {

   if(validateName(applicant.getName())==false) {
            throw new InvalidNameException("Invalid name");
   }
   if(validateAge(applicant.getAge())==false) {
    throw new InvalidAgeException("Invalid age");

   }if(validateJobProfile(applicant.getJobProfile())==false) {
    throw new InvalidJobProfileException("Invalid job profile");
   }


  }
}

class Tester {

    public static void main(String[] args) {

        try {
            Applicant applicant= new Applicant();
            applicant.setName("Jenny");
            applicant.setJobProfile("Clerk");
            applicant.setAge(25);

            Validator validator = new Validator();

            validator.validate(applicant);
            System.out.println("Application submitted successfully!");
        }
        catch (InvalidNameException|InvalidJobProfileException|InvalidAgeException e) {
            System.out.println(e.getMessage());
        }
    }
}
```