Object Oriented Techniques :-

1) Write a program to create a student class having member variable name, age, roll number. Override toString() method of the student class to print student information.

```
import java.util.Scanner;
class student {
    Scanner sc = new Scanner(System.in);
    String name;
    int age, roll;

    void set() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter name");
        name = sc.nextLine();
        System.out.println("Enter age");
        age = sc.nextInt();
        System.out.println("Enter roll");
        roll = sc.nextInt();
    }
    void ToString() {
        System.out.println("Student name" + name);
        System.out.println("Student age" + age);
        System.out.println("Student roll" + roll);
    }
}
```

```
class oops1 {
    public static void main (String[] args) {
        student student1 = new student();
        student1.set();
        student1.ToString();

        student2.set();
        student2.ToString();

        student student3 = new student();
        student3.set();
        student3.ToString();
    }
}
```

Output :-

Student name Deepak.
Student age 19.
Student roll 9.

Student name Manish.
Student age 24.
Student roll 34.

Student name Dinesh
Student age 20.
Student roll 36.

2) Override the equals of the Student class.

Here are the rules for equals() method:

It is reflexive.

x.equals(x) must be true.

It is symmetrical.

x.equals(y) is true iff y.equals(x) is true.

It is transitive.

If x.equals(y) is true & y.equals(z) is true, then x.equals(z) must be true.

It is repeatable.

Multiple calls on x.equals(y) return the same value.

It is cautios.

x.equals must return false.

```
import java.util.Scanner;
class student{
    String name;
    int age, roll, co=0;
    void set(){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter name");
        name = sc.nextLine();
        System.out.println("Enter age");
        age = sc.nextInt();
        System.out.println("Enter roll");
        roll = sc.nextInt();
        co++;
    }
```

```
void ToString(){
    System.out.println("Student name "+name);
    System.out.println("Student age "+age);
    System.out.println("Student roll "+roll);
}
void Equals(student x, student y, student z){
    if(x.equals(x))
        System.out.println("It is reflexive");
    else
        System.out.println("It is not reflexive");

    if(x.equals(y) && y.equals(x))
        System.out.println("It is symmetrical");
    else System.out.println("Not symmetrical");

    if(x.equals(y) && y.equals(z))
        if(x.equals(z))
            System.out.println("Transitive");
        else
            System.out.println("Not transitive");
    else
        System.out.println("Not transitive");

    if(x.equals(null))
        System.out.println("false");

    if(x.co>1)
        System.out.println("Not Repeatable");
    else System.out.println("Repeatable");
```

```
    }
}
class oops2 {
    public static void main (String args[]) {
        student s2 = new student();
        s1.set();
        s1.ToString();

        student s2 = new student();
        s2.set();
        s2.ToString();

        student s3 = new student();
        s3.set();
        s3.ToString();

        s1.Equals(s1,s2,s3);
    }
}
```

Output :-

```
Student name Deepak
Student age 14
Student roll 9

Student name Monish
Student age 27
Student roll 34

Student name Sailesh
Student age 20
Student roll 36
```

It is reflexive.
Not symmetrical.
Not Transitive.
It is repeatable.

3) Write a program to create a car class having member chasis number, colour, maxSpeed. Implement a user defined exception if chasis number is not a alpha numeric and number of characters is not equals to 10.

```
public class InvalidChasisNumberException extends Exception {
        InvalidChasisException () {
            super("Chasis Number is not valid");
        }
    }

public class car {
    private static String chasno;
    private static String colour;
    private static int maxSpeed;
    car (String chasno, String colour, int maxSpeed ) throws
                                    InvalidChasisException {

        this.chassno = chasno;
        this.colour = colour,
        this.maxSpeed = maxSpeed;
        Pattern P = Pattern.compile("\\P{Alnum}");
        Matcher m = p.matcher (chasno);
        if(m.find() && chasno.length()==10)
            display();
        else
    }     throw new InvalidChasisExcention();
```

```
static void display(){
        System.out.println("Car chasis no is: "+chassno);
        System.out.println("Car colour is:"+colour);
        System.out.println("Car max. speed is :"+maxSpeed);
    }
}

class main{
    public static void main(String[] args) throws
                                    InvalidChasisException{
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter chasis no:");
        String cnum = sc.nextLine();
        System.out.println("Enter car colour");
        String ccol= sc.nextLine();
        System.out.println("Enter max speed(kmph)");
        int msp = sc.nextInt();
        Car d = new Car(cnum, ccol, msp);
    }
}
```

Output :-

Car chasis no is : 12345 abcde
Car colour is : silver
Car max speed : 240

4) Create a class hierarchy having Shape class as the super class and rectangle, triangle, circle as the sub class defined their own area method. Create the reference of the super class and calculate the area of the child class object.

```
import java.util.Scanner;
class Shape {
    Scanner sc = new Scanner(System.in);
    void area() {}
}

class triangle extends Shape {
    int bore, ht;
    void area() {
    System.out.println("Enter bore :");
    bore = sc.nextInt();
    System.out.println("Enter height :");
    ht = sc.nextInt();
    System.out.println("Area of triangle " + 0.5 * bore * ht);
    }
}
class rectangle extends Shape {
    int l, b;
    void area() {
    System.out.println("Enter length :");
    l = sc.nextInt();
    System.out.println("Enter breadth :");
    b = sc.nextInt();
    System.out.println("Area of rectangle :" + l * b);
    }
}
```

```
class circle extends Shape{
    int rad;
    void area(){
    System.out.println("Enter radius");
    rad = sc.nextInt();
    System.out.println("Area of circle "+Math.PI * Math.pow
                                        (rad, 2));
    }
}

class oops4{
    public static void main(String args[]){
        Shape rectangle = new rectangle();
        rectangle.area();
        Shape triangle = new triangle();
        triangle.area();
        Shape circle = new circle();
        circle.area();
    }
}
```

Output :-

```
Enter length : 2
Enter breadth : 3
Area of rectangle is : 6

Enter base : 2
Enter height : 4
Area of triangle is : 4.0

Enter radius : 4
Area of circle : 50.26
```

5) The Integer class is one of Java's predefined Number subclasses, mentioned in the introduction to Chapter 5. It serves as a wrapper for an int value and also has static methods for parsing and formatting integers. It's fine as it is, but you may want something simpler.

Create a class, called MutableInteger, that is like an Integer but specialized by omitting the overhead of Number and providing only the set, get and incr operations.

```
import java.util.Scanner;
class MutableInteger {
    Scanner sc = new Scanner(System.in);
   int integer;
   void set(){
      System.out.println("Enter integer value");
      integer = sc.nextInt();
    }
 int get(){
     return integer;
   }
int increase(){
    return ++integer;
  }
int decrease(){
    return --integer;
  }
}
```

```
public class oops5{
    public static void main (String[] args){
        MutableInteger mt = new MutableInteger();
        mt.set();
        System.out.println(mt.get());
        System.out.println(mt.increase());
        System.out.println(mt.decrease());
    }
}
}
```

Output :-

```
Enter Integer value
 69
 69
 70
 69
```