

Practical – 1

```
E:\javaperformance\collage>javac P1_1.java
```

```
E:\javaperformance\collage>java P1_1  
Welcome to Java, Learning Java Now and Programming is fun
```

```
E:\javaperformance\collage>
```

Output : Practical – 1.1

1.1 Write a Program that displays Welcome to Java, Learning Java Now and Programming is fun.

```
class P1_1{  
    public static void main(String []args){  
        System.out.println("Welcome to Java, Learning Java Now and  
        Programming is fun");  
    }  
}
```

```
E:\javaperformance\collage>javac P1_2.java
```

```
E:\javaperformance\collage>java P1_2
```

```
Value of x is: 2.623901496861419
```

```
Value of y is: 0.7087397392563978
```

```
E:\javaperformance\collage>
```

Output : Practical – 1.2

1.2 Write a program that solves the following equation and displays the value x and y: 1) $3.4x + 50.2y = 44.5$ 2) $2.1x + .55y = 5.9$

(Assume Cramer's rule to solve equation

$$ax + by = e \quad x = \frac{ed - bf}{ad - bc}$$

$$cx + dy = f \quad y = \frac{af - ec}{ad - bc}$$

```
class P1_2{

    public static void main(String args[]){

        double a,b,c,d,e,f,x,y;

        a=3.4;
        b=50.2;
        e=44.5;
        c=2.1;
        d=0.55;
        f=5.9;

        x=(e*d - b*f)/(a*d - b*c);
        y=(a*f - e*c)/(a*d - b*c);

        System.out.println("Value of x is: "+x);
        System.out.println("Value of y is: "+y);

    }

}
```

```
E:\javaperformance\collage>javac P1_3.java
```

```
E:\javaperformance\collage>java P1_3
```

```
Enter meters :
```

```
3
```

```
Value in feet is: 9.84252
```

```
E:\javaperformance\collage>
```

Output : Practical – 1.3

1.3 Write a program that reads a number in meters, converts it to feet, and displays the result.

```
import java.util.Scanner;

class P1_3{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);

        System.out.println("Enter meters :");
        double meters=sc.nextLong();
        double feet=meters*3.28084;
        System.out.println("Value in feet is: "+feet);
    }
}
```

```
E:\javaperformance\collage>javac P1_4.java
```

```
E:\javaperformance\collage>java P1_4
```

```
Enter weight in pounds
```

```
132
```

```
Enter height in inches
```

```
68
```

```
BMI is : 20.070662978696202
```

```
E:\javaperformance\collage>
```

Output : Practical – 1.4

1.4 Body Mass Index (BMI) is a measure of health on weight. It can be calculated by taking your weight in kilograms and dividing by the square of your height in meters. Write a program that prompts the user to enter a weight in pounds and height in inches and displays the BMI.

Note:- 1 pound=.45359237 Kg and 1 inch=.0254 meters.

```
import java.util.Scanner;

class P1_4{

    public static void main(String args[]){

        Scanner sc= new Scanner(System.in);

        System.out.println("Enter weight in pounds");

        double poundWeight=sc.nextLong();

        System.out.println("Enter height in inches");

        double inchHeight=sc.nextLong();

        double kgWeight=0.4536*poundWeight;

        double meterHeight=0.0254*inchHeight;

        double userbmi=kgWeight/(meterHeight*meterHeight);

        System.out.println("BMI is : "+userbmi);

    }

}
```

```
E:\javaperformance\collage>java P1_5  
Input three integers : -5 23 9  
Decresing oredr : 23 9 -5  
E:\javaperformance\collage>
```

Output : Practical – 1.5

1.5 Write a program that prompts the user to enter three integers and display the integers in decreasing order.

```
import java.util.Scanner;

class P1_5{
    public static void main(String arg[]){
        Scanner s = new Scanner(System.in);
        int a,b,c,sm,me,la;
        System.out.print("Input three integers : ");
        a = s.nextInt(); b = s.nextInt(); c = s.nextInt();
        if(a>b){
            if(a>c){
                la = a;
                if(b>c){
                    sm = c; me = b; }
                else{
                    sm = b; me = c; }
            }
            else{
                me = a;
                if(b>c){
                    la = b; sm = c; }
                else{
                    sm = b; la = c; }
            }
        }
        else{
            if(b>c){
                la = b;
                if(c>a){
                    me = c; sm = a; }
                else{
                    me = a; sm = c; }
            }
            else{
                la = c; me = b; sm = a; }
        }
        System.out.print("Decresing oreder : ");
        System.out.println(la+" "+me+" "+sm);
    }
}
```

Practical – 2

2.1 Write a program for sorting an elements in array, and enter elements by command line argument

```
class P2_1{
    public static void main(String arg[]){
        int temp = 0;
        int a[] = new int[arg.length];
        for(int i =0 ; i < arg.length ; i++){
            a[i] = Integer.parseInt(arg[i]);

            for(int i = 0 ; i < arg.length ; i++){
                for(int j = i ; j < arg.length ; j++){
                    if(a[i] > a[j]){
                        temp = a[i];
                        a[i] = a[j];
                        a[j] = temp;
                    }
                }
            }
        }
        System.out.print("Sorted : ");
        for(int i =0 ; i < arg.length ; i++){
            System.out.print(a[i] + " ");
        }
    }
}
```

```
E:\javaperformance\collage>java P2_1 9 34 -1
Sorted : -1 9 34
E:\javaperformance\collage>
```

Output : Practical – 2.1

2.2 write a program to implement growable stack

```
import java.util.Scanner;
class P2_2{
    public static void main(String []args){
        Scanner scanf = new Scanner(System.in);
        Stack stack = new Stack();
        boolean when = true;
        int ch,num;
        System.out.printf("enter your choice \n1.push\n2.pop\n3.display\n4.Exit\n");
        while(when){
            System.out.print("enter your choice here: ");
            ch = scanf.nextInt();
            switch(ch){
                case 1: System.out.print("enter value: ");
                    num = scanf.nextInt();
                    stack.push(num);break;
                case 2: stack.pop();break;
                case 3: stack.display();break;
                case 4: when=false;break;
                default: System.out.println("Please enter valid choice");
            }
        }
    }
}

class Stack{
    private int size,top;
    private int array[]=new int[4];
    Stack(){
        top = -1;size = 4;
    }
    void push(int val){
        if(top == array.length-1) expand(val);
        else{
            top++;array[top]=val;}
    }
    void pop(){
        if(top == -1) System.out.println("UnderFlow");
        else{
```

```

        System.out.println("poped element:"+array[top]);
        top--;
    }
}
void expand(int val){
    int new_array[] = new int[size*2];
    for(int i=0;i<array.length;i++) new_array[i]=array[i];
    top=array.length;
    new_array[top]=val;
    array = new_array;
}
void display(){
    if(top== -1) System.out.println("UnderFlow");
    else{
        for(int i=0;i<=top;i++)
            System.out.println((i+1)+" element "+array[i]);
    }
}
}

```

```

E:\javaperformance\collage>java P2_2
enter your choice
1.push
2.pop
3.display
4.Exit
enter your choice here: 1
enter value: 12
enter your choice here: 1
enter value: 23
enter your choice here: 3
1 element 12
2 element 23
enter your choice here: 2
poped element:23
enter your choice here: 2
poped element:12
enter your choice here: 2
UnderFlow
enter your choice here:
4

```

Output : Practical – 2.2

2.3 Write a program that prompts the user to enter a letter and check whether a letter is a vowel or constant

```
import java.util.Scanner;
class P2_3{
    public static void main(String [] args){
        Scanner scanf = new Scanner(System.in);
        System.out.print("Enter any letter: ");
        char ch = scanf.next().charAt(0);

        if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'||ch=='A'||ch=='E'||ch=='I'||ch=='O'||ch=='U')
            System.out.println("letter is vowel");
        else
            System.out.println("letter is constant");
    }
}
```

```
E:\javaperformance\collage>java P2_3
Enter any letter: U
letter is vowel
```

```
E:\javaperformance\collage>
```

Output : Practical – 2.3

2.4 Write a program for calculator to accept an expression as a string in which the operands and operator are separated by spaces. For ex: 3 + 4

```
class P2_4{
    public static void main(String []args){
        int a = Integer.parseInt(args[0]);
        int b = Integer.parseInt(args[2]);
        switch(args[1]){
            case "+":System.out.println("Ans :"+(a+b));
            break;
            case "-":System.out.println("Ans :"+(a-b));
            break;
            case "/":System.out.println("Ans :"+(a/b));
            break;
            case "*":System.out.println("Ans :"+(a*b));
            break;
        }
    }
}
```

```
E:\javaperformance\collage>javac P2_4.java
```

```
E:\javaperformance\collage>java P2_4 4 "*" 5
```

```
Ans :20
```

Output : Practical – 2.4

Practical – 3

3.1 Write a program to perform method overloading of area method to find area of rectangle and square.

```
class P3_1{
    public static void main(String []args){
        Area rectange = new Area();
        Area square = new Area();
        rectange.Area(3,5);
        square.Area(3);
    }
}

class Area{
    int length,width;
    void Area(int a,int b){
        length=a;
        width=b;
        System.out.println("Area of rectange: "+ (width*length) );
    }
    void Area(int a){
        length=width=a;
        System.out.println("Area of square: "+ (width*length) );
    }
}
```

```
E:\javaperformance\collage\3_Practical>javac P3_1.java
```

```
E:\javaperformance\collage\3_Practical>java P3_1
```

```
Area of rectange: 15
```

```
Area of square: 9
```

Output : Practical – 3.1

3.2 Do above program using constructor overloading.

```
class P3_2{
    public static void main(String []args){
        ARea rectangle = new ARea(4,3);
        ARea square = new ARea(4);
        rectangle.Area();
        square.Area();
    }
}

class ARea{
    int length,width;
    ARea(int a,int b){
        length=a;
        width=b;
    }
    ARea(int a){
        length=width=a;
    }
    void Area(){
        System.out.println("Area: "+ (width*length) );
    }
}
```

```
E:\javaperformance\collage\3_Practical>javac P3_2.java
```

```
E:\javaperformance\collage\3_Practical>java P3_2
```

```
Area: 12
```

```
Area: 16
```

Output : Practical – 3.2

3.3 Create a class student with necessary properties, methods and constructor. Overload a function name search in this class which allows us to search student based on roll number, name and city.

```
import java.util.Scanner;
class P3_3{
    public static void main(String[] args){
        Scanner scanf = new Scanner(System.in);
        System.out.print("how many students : ");
        int number = scanf.nextInt();
        Student student = new Student(number);
        student.search(2);
        student.search("sidpro","ahmedabad");
    }
}
class Student{
    String[] name,city;
    int[] roll_num,pin_code;
    boolean found;
    Student(int num){
        name = new String[num];
        roll_num = new int[num];
        city = new String[num];
        pin_code = new int[num];
        information(num);
    }
    void information(int num){
        Scanner scanf = new Scanner(System.in);
        for(int i=0;i<num;i++){
            System.out.print("enter rollno of student "+(i+1)+": ");
            roll_num[i] = scanf.nextInt();
            scanf.nextLine();
            System.out.print("enter name of student "+(i+1)+": ");
            name[i] = scanf.nextLine();
            System.out.print("enter city of student "+(i+1)+": ");
            city[i] = scanf.nextLine();
            System.out.print("enter pincode of student "+(i+1)+": ");
            pin_code[i] = scanf.nextInt();
            scanf.nextLine();
            System.out.println();
        }
    }
    void search(int roll_number){
        found = true;
        for(int i=0;i<roll_num.length;i++){
```

```

        if(roll_num[i] == roll_number){
            System.out.print("student data found");
            System.out.printf("\n%10d%10s%10s%10d",roll_num[i],name[i],city[i],pin_code[i]);
            found = false;
        }
    }
    if(found)
        System.out.println("student data not found!");
}
void search(String Name,String City){
    found = true;
    for(int i=0;i<name.length;i++){
        if(Name.equals(name[i]) && City.equals(city[i])){
            System.out.print("student data found");
            System.out.printf("\n%10d%10s%10s%10d",roll_num[i],name[i],city[i],pin_code[i]);
            found = false;
        }
    }
    if(found)
        System.out.println("student data not found!");
}
}

```

E:\javaperformance\collage\3_Practical>java P3_3

how many students : 2

enter rollno of student 1: 30

enter name of student 1: sidpro

enter city of student 1: ahmedabad

enter pincode of student 1: 382424

enter rollno of student 2: 21

enter name of student 2: kamal

enter city of student 2: gandhinagar

enter pincode of student 2: 234098

student data not found!

student data found

30 sidpro ahmedabad 382424

Output : Practical – 3.3

Practical – 4

4.1 Write a program that create following : a inherited by b ,b inherited by c and c is inherited by d. a has one variable b has two and c has 3 variables. Use constructor and super to initialized the variables of classes.

```
class A{
    int a;
    A(int temp){
        a = temp;
    }
    void show(){
        System.out.println("class A: "+a);
    }
}
class B extends A{
    int b;
    B(int temp,int temp1){
        super(temp);
        b=temp1;
    }
    void show(){
        System.out.println("class B: "+a+" "+b);
    }
}
class C extends B{
    int c;
    C(int temp,int temp1,int temp2){
        super(temp,temp1);
        c = temp2;
    }
    void show(){
        System.out.println("class C: "+a+" "+b+" "+c);
    }
}
class D extends C{

    D(int temp,int temp1,int temp2){
        super(temp,temp1,temp2);
    }
}
```

```
    }  
    void show(){  
        System.out.println("class D: "+a+" "+b+" "+c);  
    }  
}  
class P4_1{  
    public static void main(String[] args){  
        A obj1 = new A(4);  
        B obj2 = new B(5,6);  
        C obj3 = new C(7,8,9);  
        D obj4 = new D(10,11,12);  
        obj1.show();  
        obj2.show();  
        obj3.show();  
        obj4.show();  
    }  
}
```

```
E:\javaperformance\collage\4_Practical>java P4_1  
class A: 4  
class B: 5 6  
class C: 7 8 9  
class D: 10 11 12
```

Output : Practical – 4.1

4.2 Write a Java program that combine several classes and interfaces the abstract class robot has subclass name robotA, robotB, robotC. Class robotA1 extends robotA. Class robotB1 and robotB2 extends robotB. Class robotC1 extends robotC. The locomotion interface declares three methods has forward , reverse and stop. It is implemented by class robotB and robotC. The sound interface declares one method named beep and it is implemented by robotA1,robotB1 and robotC. Defined all class and implement interface as specified than invoked beep method of all class object that are of type sound.

```
interface locomotion{
    void forward();
    void reverse();
    void stop();
}
interface sound{ void beep();}
abstract class Robot{ }
class RobotA extends Robot{ }
class RobotB extends Robot implements locomotion{
    public void forward(){
        System.out.println("Robot B is moving forward");
    }
    public void reverse(){
        System.out.println("Robot B is movinf reverse");
    }
    public void stop(){
        System.out.println("Robot B is stop");
    }
}
class RobotC extends Robot implements locomotion, sound{
    public void forward(){
        System.out.println("Robot C is moving forward");
    }
    public void reverse(){
        System.out.println("Robot C is movinf reverse");
    }
    public void stop(){
        System.out.println("Robot C is stop");
    }
    public void beep(){
        System.out.println("Robot C sounds beep.....");
    }
}
class RobotA1 extends RobotA implements sound{
    public void beep(){
        System.out.println("RobotA1 sounds beep.....");
    }
}
```

```
    }  
}  
class RobotB1 extends RobotB implements sound{  
    public void beep(){  
        System.out.println("RobotB1 sounds beep.....");  
    }  
}  
class RobotC1 extends RobotC{  
  
}  
class P4_2{  
    public static void main(String[] args){  
        RobotB objB = new RobotB();  
        RobotC objC = new RobotC();  
        RobotA1 objA1 = new RobotA1();  
        RobotB1 objB1 = new RobotB1();  
        objB.forward();  
        objB.reverse();  
        objB.stop();  
        objC.forward();  
        objC.reverse();  
        objC.stop();  
        objC.beep();  
        objA1.beep();  
        objB1.beep();  
    }  
}
```

```
E:\javaperformance\collage\4_Practical>java P4_2  
Robot B is moving forward  
Robot B is movinf reverse  
Robot B is stop  
Robot C is moving forward  
Robot C is movinf reverse  
Robot C is stop  
Robot C sounds beep.....  
RobotA1 sounds beep.....  
RobotB1 sounds beep.....
```

Output : Practical – 4.2

4.3 write a program that has abstract class dim which has a two variable dim1,dim2. triangle and rectangle are subclass of dim. implement dynamic method dispatch by creating reference of dim.

```
abstract class Dim{
    int dim1;
    int dim2;
    Dim(int a,int b){dim1=a;dim2=b;}
    abstract void show();
}
class Rectangle extends Dim{
    Rectangle(int a, int b){
        super(a,b);
    }
    void show(){
        System.out.println("Area of Rectangle:"+(dim1*dim2));
    }
}
class Triangle extends Dim{
    Triangle(int a, int b){
        super(a,b);
    }
    void show(){
        System.out.println("Area of Triangle:"+(dim1*dim2/2));
    }
}
class P4_3{
    public static void main(String[] args){
        Rectangle r = new Rectangle(4,5);
        Triangle t = new Triangle(4,6);
        Dim d;
        d = r;
        d.show();
        d = t;
        d.show();
    }
}
```

```
E:\javaperformance\collage\4_Practical>java P4_3
Area of Rectangle:20
Area of Triangle:12
```

Output : Practical – 4.3

Practical – 5

5.1 Write a java program to find solution of quadratic equation. Take care of divide by zero error and other arithmetic exceptions

```
import java.util.Scanner;
class P5_1{
    public static void main(String[] args){
        Scanner scanf = new Scanner(System.in);
        System.out.print("Enter value of a:");
        double a = scanf.nextDouble();
        System.out.print("Enter value of b:");
        double b = scanf.nextDouble();
        System.out.print("Enter value of c:");
        double c = scanf.nextDouble();
        double sq = b*b-4*a*c;
        if(sq < 0){
            System.out.println("There is no solution of quadratic equation");
        } else {
            double lo = Math.sqrt(sq);
            double x1,x2;
            sq = -1*b;x1 = sq - lo;x2 = sq + lo;
            if(a == 0) System.out.println("There is no term like this");
            else{
                x1/=(2*a);x2/=(2*a);
                System.out.println("X1: "+x1+" "+"X2: "+x2);}
        }
    }
}
```

```
E:\javaperformance\collage\5_Practical>java P5_1
Enter value of a:2
Enter value of b:-7
Enter value of c:4
X1: 0.7192235935955849 X2: 2.7807764064044154
```

Output : Practical – 5.1

5.2 Write a program in which main() call a(), a() call b(), b() call c(), c() call d(), d() generates ArrayIndexOutOfBoundsException. c and d catch throw the Exception. b() handle an Exception. Each method contain finally.

```

class P5_2{
    public static void main(String[] args){
        a();
    }
    public static void a(){
        try{
            b();
        } catch(Exception e){
            System.out.println("Now handle a(): "+e);
        }
        finally{
            System.out.println("a()'s finally is Done");
        }
    }
    public static void b(){
        try{
            c();
        } catch(Exception e){
            System.out.println("Now handle b(): "+e);
        }
        finally{
            System.out.println("b()'s finally is Done");
        }
    }
    public static void c() throws Exception{
        try{
            d();
        } catch(Exception e){
            throw new Exception("c()'s catch throw Exception");
        }
        finally{
            System.out.println("c()'s finally is Done");
        }
    }
    public static void d() throws Exception{
        try{
            throw
new
ArrayIndexOutOfBoundsException("arrayindexoutofbound");
        } catch(ArrayIndexOutOfBoundsException e){
            throw new Exception("D()'s catch throw Exception");
        }
        finally{

```

```
        System.out.println("D()'s finally is Done");  
    }  
}  
}
```

```
E:\javaperformance\collage\5_Practical>java P5_2  
D()'s finally is Done  
c()'s finally is Done  
Now handle b(): java.lang.Exception: c()'s catch throw Exception  
b()'s finally is Done  
a()'s finally is Done
```

Output : Practical – 5.2

5.3 Write a program to get value of radius through keyboard and calculate area of circle. Take care of InputMismatchException

```
import java.util.Scanner;
import java.util.InputMismatchException;
class P5_3{
    public static void main(String[] args){
        Area();
    }
    public static void Area(){
        Scanner scanf = new Scanner(System.in);
        try{
            System.out.print("Enter radius of circle: ");
            double r = scanf.nextDouble();
            double area = Math.PI * r * r;
            System.out.printf("Area of Circle: %12.2f ",area);
        } catch(InputMismatchException e){
            System.out.println("input is wrong!");
        }
    }
}
```

```
E:\javaperformance\collage\5_Practical>java P5_3
Enter radius of circle: 23
Area of Circle:      1661.90
```

Output : Practical – 5.3

5.4 Write a program to create an array of 10 integers. Get value of those 10 integers using console. Now ask for an index of array through keyboard then divide the array into two from that index. Take care of array index out of bound exception. Also handle InputMismatchException

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

class P5_4{
    public static void main(String[] args){
        Scanner scanf = new Scanner(System.in);
        int i=0,temp,j;
        int arr[] = new int[10];
        try{
            for(i=0;i<10;i++){
                temp = Integer.parseInt(args[i]);
                arr[i]=temp;
                //System.out.println(arr[i]);
            }
            System.out.println("Enter index Here: ");
            int divide = scanf.nextInt();
            int partone[] = new int[divide+1];
            int parttwo[] = new int[10-divide-1];
            for(i=0;i<=divide;++i) partone[i]=arr[i];
            for(j=0;i<10;i++,j++)parttwo[j]=arr[i];
            System.out.println("partone :");
            for(i=0;i<=divide;i++) System.out.println("    "+partone[i]);
            System.out.println("    "+"parttwo :");
            for(i=0;i<10-divide-1;i++)System.out.println("
"+parttwo[i]);

            } catch(ArrayIndexOutOfBoundsException e){
                System.out.print("Your Input is to short !");
            } catch(NegativeArraySizeException e){
                System.out.print("Your index is must be less then 10 !");
            } catch(NumberFormatException e){
                System.out.print("Your Input must contain numbers only !");
            } catch(InputMismatchException e){
                System.out.print("index is number only !");
            }
        }
    }
}
```

```
E:\javaperformance\collage\5_Practical>java P5_4 1 2 3 4 5 6 7 8 9 12
Enter index Here:
3
partone :
    1
    2
    3
    4
parttwo :
        5
        6
        7
        8
        9
        12
```

Output : Practical – 5.4

5.5 Create a class name student which stores information like roll number, name, phone number, address, course etc. Write a function which accepts an object of student to add a new student in existing list of student. While adding check for roll number. The roll number should be in 3 digit. Implement this check using user define exception class

```
import java.util.Scanner;
class Student{
    int Rollno,j;
    static int i=0;
    String Address,Course,Name,Phone;
    int rollno[] = new int[5];
    String phone[] = new String[5];
    String address[] = new String[5];
    String course[] = new String[5];
    String name[] = new String[5];

    public void AssingStudent(Student o){
        rollno[i]=o.Rollno;
        phone[i]=o.Phone;
        address[i]=o.Address;
        course[i]=o.Course;
        name[i]=o.Name;
        i++;
    }
    public void Show(){
        for(j=0;j<i;j++)
            System.out.println(rollno[j]+" "+phone[j]+" "+address[j]+" "+course[j]+"
"+name[j]);
    }
}
class P5_5{
    public static void main(String[] args){
        Student student = new Student();
        int count = 0;
        boolean loop = true;
        Scanner scanf = new Scanner(System.in);
        while(loop){
            System.out.print("Enter Roll No of Student: ");
            int temp = scanf.nextInt();scanf.nextLine();
            student.Rollno = temp;
            System.out.print("Enter Phone No of Student: ");
            student.Phone = scanf.nextLine();
            System.out.print("Enter Name of Student: ");
```



```

        student.Name = scanf.nextLine();
        System.out.print("Enter Address of Student: ");
        student.Address = scanf.nextLine();
        System.out.print("Enter Course of Student: ");
        student.Course = scanf.nextLine();
    while(temp > 0){
        temp/=10;
        count++;
    }
    try{
        if(count > 3) throw new Exception();
        count = 0;
        student.AssingStudent(student);
    } catch(Exception e){
        System.out.println("The roll number should be in 3 digit !");}
    System.out.println("You want to add new student details Y/N ?:");
    char ch = scanf.next().charAt(0);
    if(ch != 'Y' && ch != 'y') loop = false;
    }
    student.Show();
}
}

```

```

E:\javaperformance\collage\5_Practical>java P5_5
Enter Roll No of Student: 143
Enter Phone No of Student: 9929934451
Enter Name of Student: SiPro
Enter Address of Student: Ahmedabad
Enter Course of Student: CE
You want to add new student details Y/N ?:
Y
Enter Roll No of Student: 123
Enter Phone No of Student: 1234567891
Enter Name of Student: Jack
Enter Address of Student: Rajkot
Enter Course of Student: IT
You want to add new student details Y/N ?:
n
143 9929934451 Ahmedabad CE SiPro
123 1234567891 Rajkot IT Jack

```

Output : Practical – 5.5

5.6 Write the bin2Dec (string binary String) method to convert a binary string into a decimal number. Implement the bin2Dec method to throw a NumberFormatException if the string is not a binary string.

```
import java.util.Scanner;

class P5_6{
    static void bin2Dec(String check[]){
        String ch1="1",ch2="0";
        for(int i=0;i<check.length;i++){
            if(!ch1.equals(check[i]) && !ch2.equals(check[i]))
                throw new NumberFormatException("This is not binary");
        }
    }

    public static void main(String[] args){
        Scanner scanf = new Scanner(System.in);
        System.out.println("Enter Binary number");
        String bin = scanf.nextLine();
        String[] check = bin.split("");
        bin2Dec(check);
        int Dec = Integer.parseInt(bin,2);
        System.out.print("Decimal :"+Dec);
    }
}
```

```
E:\javaperformance\collage\5_Practical>java P5_6
Enter Binary number
101010101
Decimal :341
```

Output : Practical – 5.6

Practical – 6

6.1 Write a Program for multithreading that finds prime number in the range provided by user in command line argument depending upon range creates sufficient number of child thread.

```
import java.util.ArrayList;

public class P6_1 extends Thread{
    public static int temp[];
    public static int thread = 0,length=0,divide = 0;
    static ArrayList<Integer> arr = new ArrayList<>();

    public static void main(String[] args){
        int num = args.length;
        temp = new int[num];
        for( int i=0;i<num;i++){
            temp[i] = Integer.parseInt(args[i]);
        }
        length= num;
        if((num%5) == 0)divide = num/5;
        else if((num/5) == 0)divide = 1;
        else divide = (num/5)+1;

        P6_1 threaD[] = new P6_1[divide];
        for(int i=0;i<divide;i++){
            threaD[i] = new P6_1();
            threaD[i].start();
        }
        for(int i=0;i<divide;i++) {
            try {
                threaD[i].join();
            } catch (Exception e) {
                System.out.println(e);
            }
        }
        for(int i=0;i<arr.size();i++) {
            System.out.println(arr.get(i));
        }
    }

    public void run(){
        int j,count,loop = thread++;
        j=(loop)*(length/divide)+5;
        if(j > length) j = length;
    }
}
```

```
for(int i= loop*(length/divide);i<j;i++){
    //System.out.println(Thread.currentThread().getId()+" "+temp[i]);
    int num = temp[i];
    count = 0;
    if( num == 0 || num == 1)continue;
    for(int k = 2;k<=num;k++){
        if(num%k == 0) count++;
        if(count > 2)break;
    }
    if(count < 2) {
        arr.add(num);
    }
    }
}
```

```
E:\javaperformance\collage\6_Practical>java P6_1 12 13 12 1 134 34 4 45 56 6 3 5
7 8 9 10 12 23 34 44 23
Prime numbers:
13
3
5
7
7
23
23
```

Output : Practical – 6.1

6.2 Write a program that demonstrate thread priority four threads each with a different priority level then the other are started objects and not the behave of each Thread

```

class ThreaD extends Thread{
    public void run(){
        for(int i=0;i<5;i++){

            System.out.print(Thread.currentThread().getPriority()+" ");
            try{
                Thread.sleep(1000);
            } catch(Exception e){
                System.out.println(e);
            }
        }
    }
}
class P6_2 {
    public static void main(String[] args){
        ThreaD t1= new ThreaD();
        ThreaD t2= new ThreaD();
        ThreaD t3= new ThreaD();
        ThreaD t4= new ThreaD();
        ThreaD t5= new ThreaD();
        t1.setPriority(1);
        t2.setPriority(2);
        t3.setPriority(3);
        t4.setPriority(4);
        t1.start();
        t2.start();
        t3.start();
        t4.start();
        t5.start();
    }
}

```

```

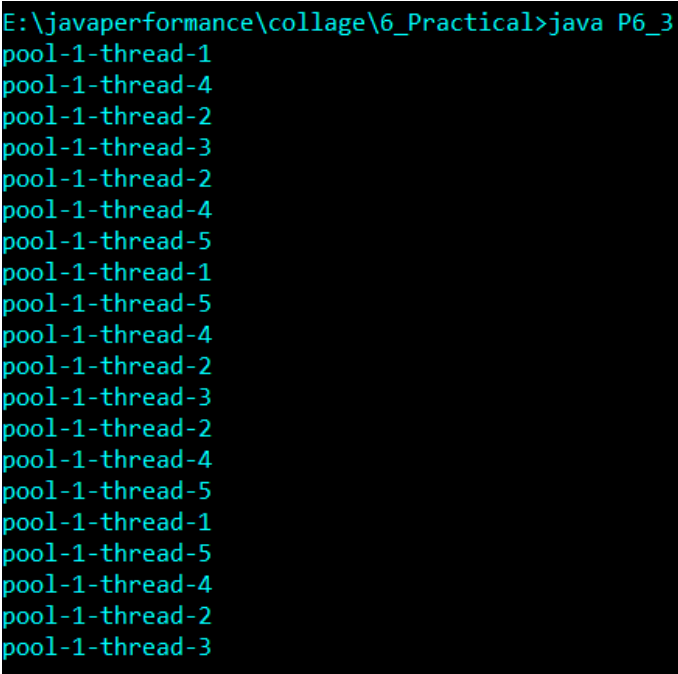
E:\javaperformance\collage\6_Practical>java P6_2
3 2 4 5 1 3 5 1 2 4 3 5 1 4 2 3 1 5 4 2 3 1 5 4 2

```

Output : Practical – 6.2

6.3 Write a program that demonstrate use of Executor Framework in multitasking.

```
import java.util.concurrent.Executors;
import java.util.concurrent.ExecutorService;
class P6_3{
    public static void main(String[] args){
        ExecutorService e = Executors.newFixedThreadPool(5);
        for(int i=0;i<20;i++){
            e.execute(new ExecutorDemo());
        }
        //System.out.println(Thread.currentThread().getName());
        e.shutdown();
    }
}
class ExecutorDemo implements Runnable{
    public void run(){
        System.out.println(Thread.currentThread().getName());
    }
}
```



```
E:\javaperformance\collage\6_Practical>java P6_3
pool-1-thread-1
pool-1-thread-4
pool-1-thread-2
pool-1-thread-3
pool-1-thread-2
pool-1-thread-4
pool-1-thread-5
pool-1-thread-1
pool-1-thread-5
pool-1-thread-4
pool-1-thread-2
pool-1-thread-3
pool-1-thread-2
pool-1-thread-4
pool-1-thread-5
pool-1-thread-1
pool-1-thread-5
pool-1-thread-4
pool-1-thread-2
pool-1-thread-3
```

Output : Practical – 6.3

6.4 Write a program for handling producer consumer problem.

```

import java.util.ArrayList;

public class ProducerConsumerDemo {
    public static void main(String[] args) {
        ArrayList<Integer> buffer = new ArrayList<>();
        Producer P = new Producer(buffer);
        Consumer C = new Consumer(buffer);
        Thread thread1 = new Thread(P);
        Thread thread2 = new Thread(C);
        thread1.start();
        thread2.start();
    }
}

class Producer implements Runnable{
    ArrayList<Integer> bufferP = new ArrayList<>();
    private int i = 0;
    public Producer(ArrayList<Integer> bufferP) {
        super();
        this.bufferP = bufferP;
    }
    public void run() {
        while(true) {
            produce(i++);
        }
    }
    public void produce(int i) {
        synchronized(bufferP) {
            while(bufferP.size() == 5) {
                System.out.println("bufferP is full ..... waiting for
consumer...");
                try {
                    bufferP.wait();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
            System.out.println("producer produce element "+i);
            bufferP.add(i);
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            bufferP.notify();
        }
    }
}

```

```

    }
}

class Consumer implements Runnable{
    ArrayList<Integer> bufferP = new ArrayList<>();

    public Consumer(ArrayList<Integer> bufferP) {
        super();
        this.bufferP = bufferP;
    }
    public void run() {
        while(true) {
            consume();
        }
    }
    public void consume() {
        synchronized(bufferP) {
            while(bufferP.isEmpty()) {
                System.out.println("bufferP is empty ..... waiting for
producer...");
                try {
                    bufferP.wait();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            System.out.println("consumed element
"+bufferP.remove(0));
            bufferP.notify();
        }
    }
}

```



```
E:\javaperformance\collage\6_Practical>java P6_4
producer produce element 0
producer produce element 1
producer produce element 2
producer produce element 3
producer produce element 4
bufferP is full ..... waiting for consumer...
consumer consumed element 0
consumer consumed element 1
consumer consumed element 2
consumer consumed element 3
consumer consumed element 4
producer produce element 5
producer produce element 6
consumer consumed element 5
producer produce element 7
producer produce element 8
producer produce element 9
producer produce element 10
bufferP is full ..... waiting for consumer...
consumer consumed element 6
consumer consumed element 7
consumer consumed element 8
producer produce element 11
```

Output : Practical – 6.4

Practical – 7

7.1 Write a program that creates an Array List and adds a Loan object , a Date object , a string, and a Circle object to the list, and use a loop to display all elements in the list by invoking the object's toString() method.

```
import java.util.ArrayList;
import java.util.Date;
import javafx.scene.shape.Circle;

class P7_1{
    public static void main(String[] args) {
        ArrayList<Object> objectlist = new ArrayList<>();
        Date d = new Date();
        objectlist.add(d);
        Circle ci = new Circle(20,40,50);
        objectlist.add(ci);
        String s = new String("Hello World!");
        objectlist.add(s);
        Loan lo = new Loan(5000,5,12,10);
        objectlist.add(lo);
        for(Object l:objectlist) {System.out.println(l.toString());}
    }
}

class Loan{
    double P,A,r,n,t;
    Loan(double P,double r,double n,double t){
        this.P = P;
        this.r = r;
        this.n = n;
        this.t = t;
    }
    void loan(){
        double temp = r/(n*100) + 1;
        A = P * Math.pow(temp,n*t);
        System.out.println("TOTAL MONEY :"+A);
    }
}
```

```
0
Wed May 27 15:01:09 IST 2020
Circle[centerX=20.0, centerY=40.0, radius=50.0, fill=0x000000ff]
Hello World!
practicefx.Loan@5cb9f472
[don, sid, lol]
```

7.2 Evaluate expression using stack class

```
import java.util.*;

public class P7_2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the expression : ");
        String s = sc.nextLine();
        System.out.println(evaluate(s));
    }
    public static int evaluate(String expr) {
        char[] array = expr.toCharArray();

        Stack<Integer> number = new Stack<Integer>();

        Stack<Character> op = new Stack<Character>();

        for (int i = 0; i < array.length; i++){
            if (array[i] == ' ')
                continue;

            if (array[i] >= '0' && array[i] <= '9')
            {
                StringBuffer sbuf = new StringBuffer();
                while (i < array.length && array[i] >= '0' && array[i] <= '9') {
                    sbuf.append(array[i++]);
                }
                number.push(Integer.parseInt(sbuf.toString()));
            }

            else if (array[i] == '(')
                op.push(array[i]);

            else if (array[i] == ')') {
                while (op.peek() != '(')
                    number.push(result(op.pop(), number.pop(), number.pop()));
                op.pop();
            }

            else if (array[i] == '+' || array[i] == '-' ||
```

```
        array[i] == '*' || array[i] == '/')
    {
        while (!op.empty() && precedence(array[i], op.peek())) {
            number.push(result(op.pop(), number.pop(), number.pop()));
        }
        op.push(array[i]);
    }
}

while (!op.empty()) {
    number.push(result(op.pop(), number.pop(), number.pop()));
}
return number.pop();
}

public static boolean precedence(char op1, char op2)
{
    if (op2 == '(' || op2 == ')')
        return false;
    if ((op1 == '*' || op1 == '/') && (op2 == '+' || op2 == '-'))
        return false;
    else
        return true;
}

public static int result(char op, int b, int a)
{
    switch (op)
    {
        case '+':
            return a + b;
        case '-':
            return a - b;
        case '*':
            return a * b;
        case '/':
            if (b == 0)
                throw new
                    UnsupportedOperationException("Cannot divide by zero");
            return a / b;
    }
}
```

```
    }  
    return 0;  
}  
}
```

```
E:\javaperformance\collage\7_Practical>java P7_2  
Enter the expression : ( 4 * 6 ) + 7  
31
```

Output : Practical – 7.2

7.3 Demonstrate test time and removed time required for hash set, tree set, an array set, linked list

```
import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;
import java.util.HashSet;
import java.util.LinkedHashSet;
import java.util.LinkedList;
import java.util.TreeSet;

public class PerformanceTest {
    static int N = 50000,i;
    public static void main(String[] args) {

        ArrayList<Integer> list = new ArrayList<>();
        for(i = 0;i < N;i++) {
            list.add(i);
        }
        Collections.shuffle(list);
        HashSet<Integer> set1 = new HashSet<>(list);
        LinkedHashSet<Integer> set2 = new LinkedHashSet<>(list);
        TreeSet<Integer> set3 = new TreeSet<>(list);
        ArrayList<Integer> list1 = new ArrayList<>(list);
        LinkedList<Integer> list2 = new LinkedList<>(list);

        System.out.println("TestTime of Hashset: "+getTestTime(set1)+" MilliSeconds");
        System.out.println("RemoveTime of HashSet: "+getRemoveTime(set1)+"
        MilliSeconds");

        System.out.println("TestTime of LinkedHashset: "+getTestTime(set2)+"
        MilliSeconds");
        System.out.println("RemoveTime of LinkedHashSet: "+getRemoveTime(set2)+"
        MilliSeconds");

        System.out.println("TestTime of Treeset: "+getTestTime(set3)+" MilliSeconds");
        System.out.println("RemoveTime of TreeSet: "+getRemoveTime(set3)+"
        MilliSeconds");

        System.out.println("TestTime of ArrayList: "+getTestTime(list1)+"
        MilliSeconds");
        System.out.println("RemoveTime of ArrayList: "+getRemoveTime(list1)+"
        MilliSeconds");
```

```
        System.out.println("TestTime of LinkedList: "+getTestTime(list2)+"
        MilliSeconds");
        System.out.println("RemoveTime of LinkedList: "+getRemoveTime(list2)+"
        MilliSeconds");
    }
    public static long getTestTime(Collection<Integer> c) {
        long StartTime = System.currentTimeMillis();

        for(int i =0;i<N;i++)
            c.contains((int)Math.random()*2*N);

        return System.currentTimeMillis() - StartTime;
    }
    public static long getRemoveTime(Collection<Integer> c) {
        long StartTime = System.currentTimeMillis();

        for(int i =0;i<N;i++)
            c.remove(i);

        return System.currentTimeMillis() - StartTime;
    }
}
```

```
TestTime of HashSet: 86 MilliSeconds
RemoveTime of HashSet: 91 MilliSeconds
TestTime of LinkedHashSet: 14 MilliSeconds
RemoveTime of LinkedHashSet: 64 MilliSeconds
TestTime of TreeSet: 26 MilliSeconds
RemoveTime of TreeSet: 136 MilliSeconds
TestTime of ArrayList: 1095 MilliSeconds
RemoveTime of ArrayList: 2250 MilliSeconds
TestTime of LinkedList: 2673 MilliSeconds
RemoveTime of LinkedList: 5776 MilliSeconds
```

Output : Practical – 7.3

7.4 Using list perform following operation on it in java program. (use ArrayList and LinkedList)

1. Creating a new list
2. Basic operations
3. Iterating over a list
4. Searching for an element in a list
5. Sorting a list
6. Copying one list into another
7. Shuffling elements in a list
8. Reversing elements in a list
9. Extracting a portion of a list
10. Converting between Lists and arrays
11. List to Stream
12. Concurrent lists.

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.ListIterator;
import java.util.Collections;
import java.util.LinkedList;

public class Listoperation {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        System.out.println("Size of Arraylist: "+list.size());
        list.add("Gabu");
        list.add("Puma");
        list.add("Manshi");
        list.add("LAMO");
        System.out.println(list.toString());
        System.out.println("Second Element of list: "+list.get(2));
        System.out.println("set at 2: "+list.set(2, "SidPro"));
        System.out.println(list.toString());
        System.out.println("Indexof Manshi: "+list.indexOf("Manshi"));
        list.add("Puma");
        System.out.println("LastIndexOf: "+list.lastIndexOf("Puma"));
        List<String> list2 = Arrays.asList("Jumo","Gabu","123","Dell");
```



```
// addall() remove() removeall() retainall()
list.addAll(list2);
System.out.println(list.toString());
list.remove(3);
System.out.println(list.toString());
list.removeAll(list2);
System.out.println(list.toString());
list.set(2, "Jumo");
list.retainAll(list2);
System.out.println(list.toString());
list.addAll(list2);

// iterator forward and backward
ListIterator<String> listiter = list.listIterator();
while(listiter.hasNext())
    System.out.print(listiter.next()+" ");
System.out.println();
ListIterator<String> listiter1 = list.listIterator(list.size());
while(listiter1.hasPrevious())
    System.out.print(listiter1.previous()+" ");

System.out.println();
// contain() and containall()
System.out.println("list contain all element of list2: "+list.containsAll(list2));
System.out.println("list contain jumo: "+list.contains("Jumo"));

// list to array
String arr[] = new String[list.size()];
list.toArray(arr);
for(String i:arr)
    System.out.print(i+" ");

// array to list
System.out.println();
ArrayList<String> fromarray = new ArrayList<>(Arrays.asList(arr));
System.out.println(fromarray.toString());

System.out.println(list.hashCode());
System.out.println("Before shuffle: "+list.toString());
Collections.shuffle(list);
System.out.println("After shuffle: "+list.toString());
Collections.sort(list);
System.out.println("After sorting: "+list.toString());

// LINKED LIST here
```

```
LinkedList<String> link = new LinkedList<>(list);
System.out.println("Linked list: "+link.toString());
link.addFirst("Word");
link.addLast("XYZ");
System.out.println(link.toString());
System.out.println("First element of link: "+link.getFirst());
System.out.println("Last element of link: "+link.getLast());
link.removeFirst();
link.removeLast();
System.out.println("After removing first and last: "+link.toString());

System.out.println("Display the linked list forward:");
ListIterator<String> listIterator = link.listIterator();
while (listIterator.hasNext()) {
    System.out.print(listIterator.next() + " ");
}
System.out.println();

System.out.println("Display the linked list backward:");
listIterator = link.listIterator(link.size());
while (listIterator.hasPrevious()) {
    System.out.print(listIterator.previous() + " ");
}
System.out.println();
link.remove(1);
System.out.println(link.toString());
System.out.println("link contains Gabu: "+link.contains("Gabu"));
System.out.println("last index of Jumo: "+link.lastIndexOf("Jumo"));
System.out.println("first index of Jumo: "+link.indexOf("Jumo"));

Collections.shuffle(link);
System.out.println(link.toString());
Collections.sort(link);
System.out.println(link.toString());
link.removeLastOccurrence("Jumo");
System.out.println(link.toString());
}
}
```

```
Size of ArrayList: 0
[Gabu, Puma, Manshi, LAMO]
Second Element of list: Manshi
set at 2: Manshi
[Gabu, Puma, SidPro, LAMO]
Indexof Manshi: -1
LastIndexOf: 4
[Gabu, Puma, SidPro, LAMO, Puma, Jumo, Gabu, 123, Dell]
[Gabu, Puma, SidPro, Puma, Jumo, Gabu, 123, Dell]
[Puma, SidPro, Puma]
[Jumo]
Jumo Jumo Gabu 123 Dell
Dell 123 Gabu Jumo Jumo
list contain all element of list2: true
list contain jumo: true
Jumo Jumo Gabu 123 Dell
[Jumo, Jumo, Gabu, 123, Dell]
-1922259845
Before shuffle: [Jumo, Jumo, Gabu, 123, Dell]
After shuffle: [Jumo, Gabu, 123, Dell, Jumo]
After sorting: [123, Dell, Gabu, Jumo, Jumo]
Linked list: [123, Dell, Gabu, Jumo, Jumo]
[Word, 123, Dell, Gabu, Jumo, Jumo, XYZ]
First element of link: Word
Last element of link: XYZ
After removing first and last: [123, Dell, Gabu, Jumo, Jumo]
Display the linked list forward:
123 Dell Gabu Jumo Jumo
Display the linked list backward:
Jumo Jumo Gabu Dell 123
[123, Gabu, Jumo, Jumo]
link contains Gabu: true
```

Output : Practical – 7.4

7.5 Write a java program to evaluate arithmetic operation using stack

```
import java.util.*;

public class P7_5 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the expression : ");
        String s = sc.nextLine();
        System.out.println(evaluate(s));
    }
    public static int evaluate(String expr) {
        char[] array = expr.toCharArray();

        Stack<Integer> number = new Stack<Integer>();

        Stack<Character> op = new Stack<Character>();

        for (int i = 0; i < array.length; i++){
            if (array[i] == ' ')
                continue;

            if (array[i] >= '0' && array[i] <= '9')
            {
                StringBuffer sbuf = new StringBuffer();
                while (i < array.length && array[i] >= '0' && array[i] <= '9') {
                    sbuf.append(array[i++]);
                }
                number.push(Integer.parseInt(sbuf.toString()));
            }

            else if (array[i] == '(')
                op.push(array[i]);

            else if (array[i] == ')') {
                while (op.peek() != '(')
                    number.push(result(op.pop(), number.pop(), number.pop()));
                op.pop();
            }

            else if (array[i] == '+' || array[i] == '-' ||
```

```
        array[i] == '*' || array[i] == '/')
    {
        while (!op.empty() && precedence(array[i], op.peek())) {
            number.push(result(op.pop(), number.pop(), number.pop()));
        }
        op.push(array[i]);
    }
}

while (!op.empty()) {
    number.push(result(op.pop(), number.pop(), number.pop()));
}
return number.pop();
}

public static boolean precedence(char op1, char op2)
{
    if (op2 == '(' || op2 == ')')
        return false;
    if ((op1 == '*' || op1 == '/') && (op2 == '+' || op2 == '-'))
        return false;
    else
        return true;
}

public static int result(char op, int b, int a)
{
    switch (op)
    {
        case '+':
            return a + b;
        case '-':
            return a - b;
        case '*':
            return a * b;
        case '/':
            if (b == 0)
                throw new
                    UnsupportedOperationException("Cannot divide by zero");
            return a / b;
    }
}
```

```
    }  
    return 0;  
}  
}
```

```
E:\javaperformance\collage\7_Practical>java P7_2  
Enter the expression : ( 4 * 6 ) + 7  
31
```

Output : Practical – 7.5

7.6 Implement a java program to show various operation of queue

```
import java.util.LinkedList;
import java.util.Queue;

public class Queueoperation {
    public static void main(String[] args) {
        Queue<String> queue = new LinkedList<>();
        queue.offer("Sidpro");
        queue.offer("Pro");
        queue.offer("Lamo");
        queue.offer("Gabu");
        queue.offer("Dhuni");
        System.out.println("queue: "+queue.toString());
        System.out.println("peek value: "+queue.peek());
        System.out.println("poll out peek value: "+queue.poll());
        System.out.println("After polling: "+queue.toString());
    }
}
```

```
queue: [Sidpro, Pro, Lamo, Gabu, Dhuni]
peek value: Sidpro
poll out peek value: Sidpro
After polling: [Pro, Lamo, Gabu, Dhuni]
```

Output : Practical – 7.6

Practical – 8

8.1 Implement singly linked list and its operations in java program

```
import java.util.ListIterator;
import java.util.Collections;
import java.util.LinkedList;

public class Listoperation {
    public static void main(String[] args) {
        LinkedList<String> link = new LinkedList<>(list);
        System.out.println("Linked list: "+link.toString());
        link.addFirst("Word");
        link.addLast("XYZ");
        System.out.println(link.toString());
        System.out.println("First element of link: "+link.getFirst());
        System.out.println("Last element of link: "+link.getLast());
        link.removeFirst();
        link.removeLast();
        System.out.println("After removing first and last: "+link.toString());

        System.out.println("Display the linked list forward:");
        ListIterator<String> listIterator = link.listIterator();
        while (listIterator.hasNext()) {
            System.out.print(listIterator.next() + " ");
        }
        System.out.println();

        System.out.println("Display the linked list backward:");
        listIterator = link.listIterator(link.size());
        while (listIterator.hasPrevious()) {
            System.out.print(listIterator.previous() + " ");
        }
        System.out.println();
        link.remove(1);
        System.out.println(link.toString());
        System.out.println("link contains Gabu: "+link.contains("Gabu"));
        System.out.println("last index of Jumo: "+link.lastIndexOf("Jumo"));
        System.out.println("first index of Jumo: "+link.indexOf("Jumo"));
```



```
Collections.shuffle(link);
System.out.println(link.toString());
Collections.sort(link);
System.out.println(link.toString());
link.removeLastOccurrence("Jumo");
System.out.println(link.toString());
}
}
```

```
Linked list: [123, Dell, Gabu, Jumo, Jumo]
[Word, 123, Dell, Gabu, Jumo, Jumo, XYZ]
First element of link: Word
Last element of link: XYZ
After removing first and last: [123, Dell, Gabu, Jumo, Jumo]
Display the linked list forward:
123 Dell Gabu Jumo Jumo
Display the linked list backward:
Jumo Jumo Gabu Dell 123
[123, Gabu, Jumo, Jumo]
link contains Gabu: true
last index of Jumo: 3
first index of Jumo: 2
[Jumo, Jumo, 123, Gabu]
[123, Gabu, Jumo, Jumo]
[123, Gabu, Jumo]
```

Output : Practical – 8.1

8.2 Define MyPriorityQueue class that extends Priority Queue to implement the Cloneable interface and implement the clone() method to clone a priority queue

```
import java.util.*;
public class MyPriorityQueue extends PriorityQueue implements Cloneable{
    public Object clone()throws CloneNotSupportedException{
        return (MyPriorityQueue)super.clone();
    }
    public static void main(String arg[]) throws CloneNotSupportedException{
        MyPriorityQueue Que= new MyPriorityQueue();
        Que.add(10);
        Que.add(20);
        Object Que2 = Que.clone();
        System.out.println(Que);
        System.out.println(Que2);
    }
}
```

```
E:\javaperformance\collage\7_Practical>java MyPriorityQueue
[10, 20]
[10, 20]
```

Output : Practical – 8.2

8.3.1 Write a Java program to remove all occurrences of a specified value in a given array of integers and return the new length of the array.

```
import java.util.*;
public class P8_3_1{
    public static void main(String arg[]){
        Scanner sc = new Scanner(System.in);
        int A[]={ 1,1,2,3,4,7,9,12,40};
        int n = sc.nextInt();
        ArrayList<Integer> Anew = new ArrayList<>();
        for(int i = 0; i < A.length; i++){
            if(A[i]!=n)
                Anew.add(A[i]);
        }
        System.out.println(Anew.toString());
    }
}
```

```
E:\javaperformance\collage\8_Practical>javac P8_3_1.java
E:\javaperformance\collage\8_Practical>java P8_3_1
1
[2, 3, 4, 7, 9, 12, 40]
```

8.3.2 Write a Java program to find the longest word in a text file

```
import java.io.File;
import java.io.IOException;
import java.util.Scanner;

public class P8_3_2 {
    public static void main(String[] args) throws IOException {
        File file = new File("E://javaperformance//Collage//8_Practical//exp.txt");
        Scanner scanf = new Scanner(file);
        String string,max = "";
        while(scanf.hasNext()) {
            string = scanf.next();
            if(string.length() > max.length())
                max = string;
        }
        System.out.println(max);
        scanf.close();
    }
}
```

```
E:\javaperformance\collage\8_Practical>java P8_3_2
helth
```

Output : Practical – 8.3

8.4 Write a program to create a file name 123.txt, if it does not exist. Append a new data to it if it already exist. write 150 integers created randomly into the file using Text I/O. Integers are separated by space.

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Scanner;

public class P8_4 {
    public static void main(String[] args) throws IOException {
        File file = new File("E://javaperformance//Collage//8_Practical//123.txt");
        // if(file.exists()) {
        //     System.out.println("File already exist");
        //     System.exit(0);
        // }
        PrintWriter write = new PrintWriter(new FileWriter(file,true));
        for(int i = 0;i<150;i++) {
            int num = i;
            write.print(num+" ");
        }
        write.close();
        int i = 0;
        Scanner scanf = new Scanner(new
        File("E://javaperformance//Collage//8_Practical//123.txt"));
        while(scanf.hasNext()) {
            int num = scanf.nextInt();
            System.out.print(num + " ");
            i++;
            if(i%10 == 0)System.out.println();
        }
        scanf.close();
    }
}
```

```
E:\javaperformance\collage\8_Practical>java P8_4
0 1 2 3 4 5 6 7 8 9
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49
50 51 52 53 54 55 56 57 58 59
60 61 62 63 64 65 66 67 68 69
70 71 72 73 74 75 76 77 78 79
80 81 82 83 84 85 86 87 88 89
90 91 92 93 94 95 96 97 98 99
100 101 102 103 104 105 106 107 108 109
110 111 112 113 114 115 116 117 118 119
120 121 122 123 124 125 126 127 128 129
130 131 132 133 134 135 136 137 138 139
140 141 142 143 144 145 146 147 148 149
```

8.5 Write a program that reads words from a text file and displays all the nonduplicate words in descending order. The text file is passed as a command-line argument

```
import java.io.File;
import java.io.IOException;
import java.util.Iterator;
import java.util.Scanner;

import java.util.TreeSet;

public class P8_5 {
    public static void main(String[] args) throws IOException {
        TreeSet<String> set = new TreeSet<>();
        File file = new File("E://javaperformance//Collage//8_Practical//" + args[0]);
        Scanner scanf = new Scanner(file);
        System.out.println("Total words: ");
        while(scanf.hasNext()) {
            String temp = scanf.nextLine();
            System.out.print(temp + " ");
            set.add(temp);
        }
        //System.out.print(set);
        System.out.println();
        Iterator<String> des = set.descendingIterator();
        System.out.println("nonduplicate words in descending order:");
        while(des.hasNext()) {
            System.out.print(des.next()+" ");
        }
        scanf.close();
    }
}
```

```
E:\javaperformance\collage\8_Practical>java P8_5 exp.txt
Total words:
lol LOL pop push rom sid pro gabu tunu rom tunu cj helth back black cj
nonduplicate words in descending order:
tunu sid rom push pro pop lol helth gabu cj black back LOL
```

Output : Practical – 8.5

Practical – 9

9.1 Write a program that moves a circle up, down, left or right using arrow keys.

```
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.control.Button;
import javafx.scene.input.KeyEvent;
//import javafx.scene.layout.StackPane;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.scene.shape.Circle;
import javafx.scene.paint.Color;
public class JavafxMoveCircle extends Application{
    public static void main(String[] args) {
        launch(args);
    }
    @Override
    public void start(Stage primaryStage) throws Exception {
        Button btn = new Button("click");
        Button exit = new Button("EXIT");
        exit.setOnAction(e ->{
            System.out.println("Now you are leaving");
            System.exit(0);
        });
        btn.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                System.out.println("Hello Sidpro");
            }
        });
        //define everything
        Circle circle = new Circle();
        Stage stage = new Stage();
        Pane pane1 = new Pane();
        pane1.setStyle("-fx-background-color: lightblue");
        VBox pane = new VBox();

        // circle is here
        circle.setCenterX(50);
        circle.setCenterY(50);
        //circle.centerXProperty().bind(pane1.widthProperty().divide(2));
        //circle.centerYProperty().bind(pane1.heightProperty().divide(2));
```

```

        circle.setRadius(25);
        circle.setStroke(Color.BLACK);
        circle.setFill(Color.GREEN);

        // stage with circle
        pane1.getChildren().add(circle);
        pane1.setMaxWidth(500);
        Scene scene1 = new Scene(pane1,500,500);
        stage.setTitle("stage 2");
        stage.setScene(scene1);
        stage.show();
        stage.setResizable(false);

        //pane1 event
        scene1.setOnKeyPressed(new EventHandler<KeyEvent>() {
            public void handle(KeyEvent event) {
                switch(event.getCode()) {
                    case UP:      circle.setCenterX(circle.getCenterX());

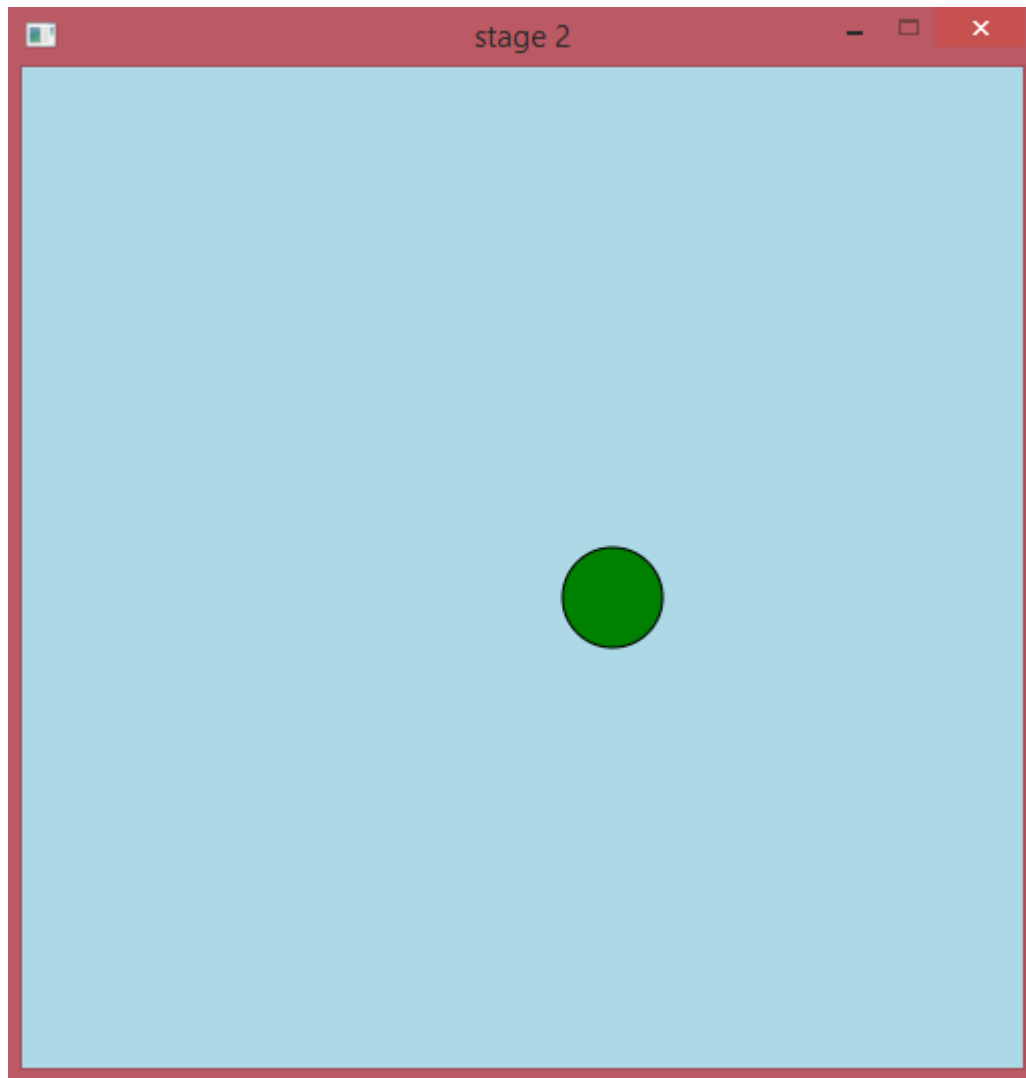
circle.setCenterY(Math.max(25,circle.getCenterY()- 10));
                                break;
                    case DOWN:  circle.setCenterX(circle.getCenterX());

circle.setCenterY(Math.min(475,circle.getCenterY()+ 10));
                                break;
                    case LEFT:  circle.setCenterX(Math.max(25,circle.getCenterX() -
10));
                                circle.setCenterY(circle.getCenterY());
                                break;
                    case RIGHT: circle.setCenterX(Math.min(475,circle.getCenterX()
+ 10));
                                circle.setCenterY(circle.getCenterY());
                                break;
                    default:    System.out.println("WRONG KEY");
                                break;
                }
            }
        });

        // primary stage
        pane.getChildren().addAll(btn,exit);
        Scene scene = new Scene(pane,200,200,Color.BEIGE);
        primarystage.setTitle("HOOM");
        primarystage.setScene(scene);
        primarystage.show();

```

```
        primarystage.setResizable(false);  
    }  
}
```



Output : Practical – 9.1

9.2 Write a program that displays the color of a circle as red when the mouse button is pressed and as blue when the mouse button is released.

```
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.control.Button;
import javafx.scene.layout.GridPane;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.stage.Stage;

public class JavaFxRedGreen extends Application {
    public static void main(String[] args) {
        launch(args);
    }

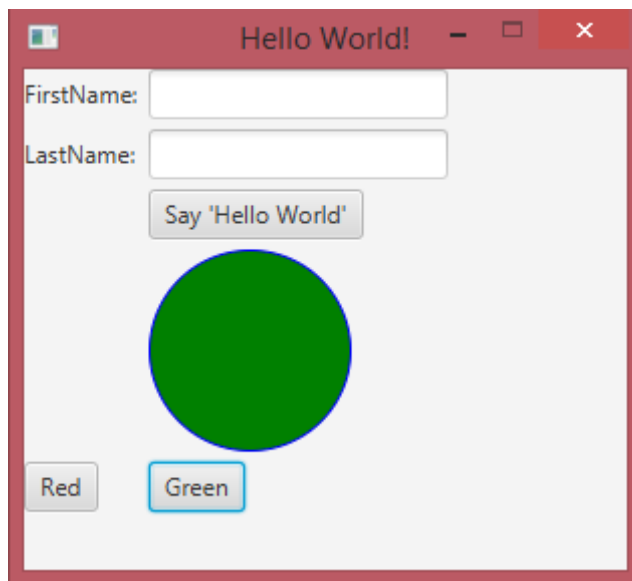
    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Hello World!");
        Button btn = new Button();
        btn.setText("Say 'Hello World'");
        btn.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent event) {
                System.out.println("Hello World!");
            }
        });
        Button Red = new Button("Red");
        Button Green = new Button("Green");

        Label label1 = new Label("FirstName:");
        TextField field1 = new TextField();
        Label label2 = new Label("LastName:");
        TextField field2 = new TextField();

        Circle circle = new Circle();
        circle.setRadius(50);
        circle.setFill(Color.WHITE);
        circle.setStroke(Color.BLUE);
```

```
Red.setOnAction(e ->{
    circle.setFill(Color.RED);
});
Green.setOnAction(e -> {
    circle.setFill(Color.GREEN);
});
GridPane gridpane = new GridPane();
gridpane.setHgap(5);
gridpane.setVgap(5);
gridpane.add(label1, 0, 0);
gridpane.add(field1, 1, 0);
gridpane.add(label2, 0, 1);
gridpane.add(field2, 1, 1);
gridpane.add(btn, 1, 2);
gridpane.add(circle, 1, 3);
gridpane.add(Red, 0, 4);
gridpane.add(Green, 1, 4);
primaryStage.setScene(new Scene(gridpane, 300, 250));
primaryStage.show();
primaryStage.setResizable(false);
}
}
```



Output : Practical – 9.2

9.3 Write a GUI program that use button to move the message to the left and right and use the radio button to change the color for the message displayed.

```
import javafx.application.Application;
import javafx.event.EventHandler;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.RadioButton;
import javafx.scene.control.ToggleGroup;
import javafx.scene.input.KeyEvent;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.Pane;
import javafx.scene.layout.VBox;
import javafx.scene.paint.Color;

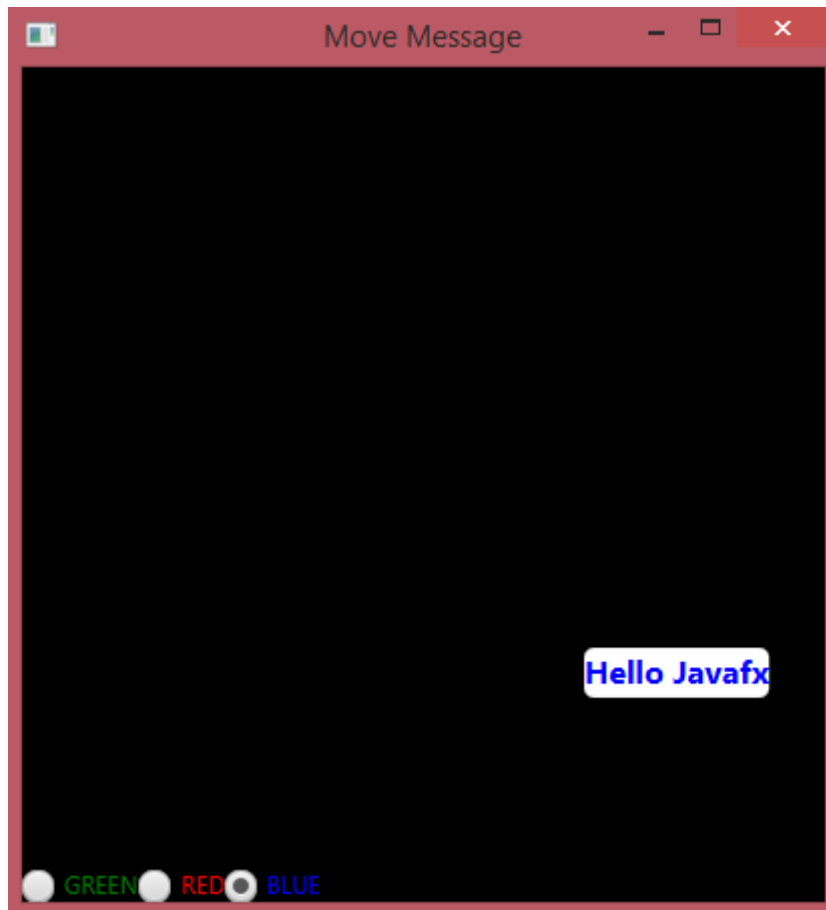
public class MoveMessage extends Application{
    public static void main(String[] args) {
        launch(args);
    }
    public void start(Stage stage) throws Exception {
        stage.setTitle("Move Message");
        Label text = new Label("Hello Javafx");
        text.setStyle("-fx-background-color:white;-fx-background-radius:5px;-fx-font-size:16px;-fx-font-weight:bold");
        ToggleGroup gp = new ToggleGroup();
        RadioButton radiored = new RadioButton("RED");
        radiored.setTextFill(Color.RED);
        radiored.setToggleGroup(gp);
        radiored.setSelected(false);
        radiored.setFocusTraversable(false);
        radiored.setStyle("-fx-focus-color:transparent;-fx-faint-focus-color:transparent");
        RadioButton radiogreen = new RadioButton("GREEN");
        radiogreen.setTextFill(Color.GREEN);
        radiogreen.setToggleGroup(gp);
        radiogreen.setFocusTraversable(false);
        radiogreen.setStyle("-fx-focus-color:transparent;-fx-faint-focus-color:transparent");
        RadioButton radioyellow = new RadioButton("BLUE");
        radioyellow.setTextFill(Color.BLUE);
        radioyellow.setToggleGroup(gp);
        radioyellow.setFocusTraversable(false);
        radioyellow.setStyle("-fx-focus-color:transparent;-fx-faint-focus-color:transparent");

        radiored.setOnAction( e -> {
            if(radiored.isSelected()) {
                text.setTextFill(Color.RED);
            }
        });
    }
}
```

```
        }
    });
    radiogreen.setOnAction(e -> {
        if(radiogreen.isSelected()) {
            text.setTextFill(Color.GREEN);
        }
    });
    radioyellow.setOnAction(e -> {
        if(radioyellow.isSelected()) {
            text.setTextFill(Color.BLUE);
        }
    });
    GridPane gridpane = new GridPane();
    gridpane.add(radiogreen, 0, 0);
    gridpane.add(radiored, 1, 0);
    gridpane.add(radioyellow, 2, 0);
    gridpane.setStyle("-fx-background-color:black");
    Pane pane = new Pane();
    pane.setStyle("-fx-background-color:black");
    pane.getChildren().add(text);
    pane.setPrefHeight(400);
    pane.setPrefWidth(400);
    VBox vbox = new VBox();
    vbox.getChildren().addAll(pane,gridpane);

    Scene scene = new Scene(vbox,400,417);
    pane.requestFocus();
    scene.setOnKeyPressed(new EventHandler<KeyEvent>() {
        public void handle(KeyEvent event) {
            switch(event.getCode()) {
                case UP:      text.setLayoutY(Math.max(0,text.getLayoutY() - 10));
                             break;
                case DOWN:   text.setLayoutY(Math.min(375,text.getLayoutY() + 10));
                             break;
                case LEFT:   text.setLayoutX(Math.max(0,text.getLayoutX() - 10));
                             break;
                case RIGHT:  text.setLayoutX(Math.min(308,text.getLayoutX() + 10));
                             break;
                default:      System.out.println("WRONG KEY");
                             break;
            }
        }
    });
    stage.setScene(scene);
    stage.show();
}
```

```
}
```



Output : Practical – 9.3