

1. Develop programs on Control Structures and Type Conversions in java.

Aim : To develop a java program to find grades of student by accepting marks.

Program :

```
import java.util.*;

class GradeOfAStudent
{
    public static void main(String agrs[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter a Average of a Student : ");
        int n=sc.nextInt();
        if( n>90 )
            System.out.println("A+ Grade");
        else if(n>80 && n<=90)
            System.out.println("A Grade");
        else if(n>70 && n<=80)
            System.out.println("B Grade");
        else if(n>60 && n<=70)
            System.out.println("C Grade");
        else if(n>50 && n<=60)
            System.out.println("D Grade");
        else if(n>40 && n<=50)
            System.out.println("E Grade");
        else
            System.out.println("Fail") ;
    }
}
```

OUTPUT:

Enter a Average of a Student : 60

D Grade

Aim : To develop a java program to find factorial of given number.

Program :

```
import java.util.*;

class Factorial

{

    public static void main(String args[])

    {

        Scanner sc=new Scanner(System.in);

        System.out.print("Enter a Number : ");

        int n=sc.nextInt(),f=1;

        for(int i=1;i<=n;i++)

            f=f*i;

        System.out.println("Factorial number of "+n+" is "+f);

    }

}
```

Output:

Enter a Number : 5

Factorial number of 5 is 120

Aim : To develop a java program to check given number is prime or not.

Program :

```
import java.util.*;

class PrimeOrNot {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a Number : ");

        int n = sc.nextInt(), flag = 0;

        for (int i = 2; i <= (n / 2); i++) {

            if (n % i == 0)

                flag = 1;

            break;

        }

    }

}
```

```

    }
    if (flag == 0)
        System.out.println(n + " is a Prime Number");
    else
        System.out.println(n + " is not a Prime Number");
    }
}

```

Output:

Enter a Number : 2
2 is a Prime Number

Aim : To develop a java program to check given number is palindrome or not.

Program :

```

import java.util.*;

class PalindromeOrNot{

    public static void main(String args[]){

        Scanner sc=new Scanner(System.in);

        System.out.print("Enter a Number : ");

        int n=sc.nextInt(),N=n,a=0;

        while(n>0){

            a=a *10+( n% 10);

            n=n/10;

        }

        if (N==s)

            System.out.println(N+" is a Palindrome");

        else

            System.out.println(N+" is not a Palindrome");

    }

}

```

Output-1:

Enter a Number : 1234321

1234321 is a Palindrome

Output-2

Enter a Number : 112345

112345 is not a Palindrome

Aim : To develop a java program to find prime numbers in the given range.

Program :

```
import java.util.*;

class PrimeRange {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("enter m value : ");

        int m = sc.nextInt();

        System.out.println("enter n value : ");

        int n = sc.nextInt(), i = m, j;

        System.out.println("Prime numbers are : ");

        if (m < 2)

            i = 2;

        while (i <= n) {

            j = 2;

            while (j <= i / 2) {

                if ((i % j) == 0)

                    break;

                j++;

            }

            if (j > i/2)

                System.out.print(i + " ");

            i++;

        }

    }

}
```

Output:

enter m value :

1

enter n value :

20

Prime numbers are :

2 3 5 7 11 13 17 19

Aim : To develop a java program to illustrate type conversions.

Program :

```
import java.util.*;

class TypeConversionAndCasting {
    public static void main(String args[]) {
        int i1 = 3;
        float f1 = i1; //Type conversion
        System.out.println(i1+" Integer");
        System.out.println(f1+" Float");
        float f2 = 10.0f;
        int i2 = (int) f2; // Type casting
        System.out.println( f2+ " Float");
        System.out.println(i2 + " Integer");
        char c=(char) i1; //integer to character
        System.out.println(c+" Character");
    }
}
```

Output:

3 Integer

3.0 Float

10.0 Float

10 Integer

♥ Character

2. Develop programs using various String handling functions

Aim : To develop a java program to check given number is palindrome or not.

Program :

```
import java.util.*;
class Palindrome {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the String :");
        String s = sc.next();
        s = s.toLowerCase();
        int n = s.length() - 1, i = 0;
        while ((i < n / 2) && (s.charAt(i) == (s.charAt(n - i))))
            i++;
        if (i == n / 2)
            System.out.println("Palindrome");
        else
            System.out.println("Not a Palindrome");
    }
}
```

Output 1:

Enter the String :

Madam

Palindrome

Output 2:

Enter the String :

Sir

Not a Palindrome

Aim : To develop a java program to find no of uppercase ,lower case ,vowels ,consonants and special characters and numbers in a given string.

Program :

```
import java.util.*;
class CountChar {
    static boolean testVowel(char ch) {
        int i;
        char a[] = { 'a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U' };
    }
}
```

```

    for (i = 0; i < a.length; i++) {
        if (ch == a[i])
            return true;
    }
    return false;
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    String s;
    System.out.println("enter the String :");
    s = sc.nextLine();
    int lc = 0, uc = 0, nc = 0, vc = 0, cc = 0, spc = 0, i;
    char c;
    for (i = 0; i < s.length(); i++) {
        c = s.charAt(i);
        if (c >= 'A' && c <= 'Z')
            uc++;
        else if (c >= 'a' && c <= 'z')
            lc++;
        else if (c >= '0' && c <= '9')
            nc++;
        else
            spc++;
        if (testVowel(c))
            vc++;
        else if (c >= 'A' && c <= 'z')
            cc++;
    }
    System.out.println("lowercase : " + lc);
    System.out.println("uppercase : " + uc);
    System.out.println("integers : " + nc);
    System.out.println("vowels : " + vc);

```

```
        System.out.println("consonants : " + cc);  
        System.out.println("special characters : " + spc);  
    }  
}
```

Output:

RajuRaviSurya@Btech_CAI@21A81A43__

lowercase : 14

uppercase : 9

integers : 6

vowels : 11

consonants : 15

special characters : 5

3. Construct programs using the following concepts:

a) Classes & Objects b) Usage of static c) Constructors

Aim : To develop a java program to create student class to read and display of student details.

Program :

```
import java.util.*;
class Std {
    String rno;
    String name;
    float cgpa;
    Scanner sc = new Scanner(System.in);
    void read() {
        System.out.println("Enter Student id :");
        rno = sc.next();
        System.out.println("Enter Student name :");
        name = sc.next();
        System.out.println("Enter Student cgpa :");
        cgpa = sc.nextFloat();
    }
    void disp() {
        System.out.println("rno : " + rno);
        System.out.println("name : " + name);
        System.out.println("cgpa : " + cgpa);
    }
}
class StdDemo {
    public static void main(String args[]) {
        Std s = new Std();
        s.read();
        System.out.println("-----");
        s.disp();
        System.out.println("-----");
    }
}
```

Output:

Enter Student id :

21A81A43__

Enter Student name :

Raju

Enter Student cgpa :

9.33

rno : 21A81A43__

name : Raju

cgpa : 9.33

Aim : To develop a java program to illustrate array of objects for student class.

Program :

```
import java.util.*;

class StdArray {

    String rno;

    String name;

    float cgpa;

    Scanner sc = new Scanner(System.in);

    StdArray () {

        System.out.println("Enter Student id :");

        rno = sc.next();

        System.out.println("Enter Student name :");

        name = sc.next();

        System.out.println("Enter Student cgpa :");

        cgpa = sc.nextFloat();

    }

    void disp() {

        System.out.println("rno : " + rno);

        System.out.println("name : " + name);

        System.out.println("cgpa : " + cgpa);

    }

}

class StdArrayMain {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter no of Students :");

        int n = sc.nextInt(),i;

        StdArray s[] = new StdArray[n];
```

```

    for (i = 0; i < n; i++) {
        s[i] = new StdArray();
    }
    System.out.println("-----");
    for (i = 0; i < n; i++) {
        s[i].disp();
        System.out.println("-----");
    }
}
}
}

```

Output:

Enter no of Students :

3

Enter Student id :

21a81a41

Enter Student name :

Raju

Enter Student cgpa :

9.23

Enter Student id :

21a81a42

Enter Student name :

Priya

Enter Student cgpa :

9.25

Enter Student id :

21a81a43

Enter Student name :

Ravi

Enter Student cgpa :

8.90

rno : 21a81a41

name : Raju

cgpa : 9.33

rno : 21a81a42

name : Priya

cgpa : 9.25

rno : 21a81a43

name : Ravi

cgpa : 8.9

Aim : To develop a java program to illustrate static variables and static methods.

Program :

```
class StaticTest {
    static int a = 62;
    static int b = 64;
    static void disp() {
        System.out.println("Static Method Display ");
    }
}
class StaticDemo {
    public static void main(String args[]) {
        System.out.println(StaticTest.a+"    "+StaticTest.b);
        StaticTest.disp();
    }
}
```

Output:

62 64

Static Method Display

Aim : To develop a java program to illustrate default and parameter constructor.

Program :

```
class DefAndPara {
    String s;
    DefAndPara() {
        System.out.println("Default Constructor");
    }
    DefAndPara(String s) {
        this.s=s;
        System.out.println(s);
        System.out.println("Parameter Constructor");
    }
}
class DefaultAndParameterConstructor {
    public static void main(String[] args) {
        DefAndPara d = new DefAndPara();
        DefAndPara p = new DefAndPara("Hello ,World!");
    }
}
```

Output :

Default Constructor

Hello ,World!

Parameter Constructor

Aim : To develop a java program to illustrate constructor overloading.

Program :

```
class COTest{
    String s;
    int i;
    COTest() {
        System.out.println("Default constructor");
    }
    COTest(String s) {
        this.s = s;
        System.out.println("String constructor");
    }
    COTest(int i) {
```

```

        this.i = i;

        System.out.println("Integer constructor");
    }
}

class CODemo {

    public static void main(String[] args) {

        COTest s1 = new COTest();

        COTest s2 = new COTest(10);

        COTest s3 = new COTest("suresh");

    }
}

```

Output:

Default constructor

Integer constructor

String constructor

Aim : To develop a java program to create copy of an object using constructor.

Program :

```

class CopyConstructorTest{

    String s;

    int i;

    CopyConstructorTest() {

        s = "Raju";

        i = 3333;

    }

    CopyConstructorTest(CopyConstructorTest x) {

        this.s = x.s;

        this.i = x.i;

    }

}

class CopyConstructor {

```

```
public static void main(String[] args) {  
    CopyConstructorTest s1 = new CopyConstructorTest();  
    CopyConstructorTest s2 = new CopyConstructorTest(s1);  
    System.out.println(" S1 values : " + s1.s + " " + s1.i);  
    System.out.println(" S2 values : " + s2.s + " " + s2.i);  
}  
}
```

Output:

S1 values : Raju 3333

S2 values : Raju 3333

4. Construct programs using the following concepts.

a) Arrays b)Nested Classes c) Command Line Arguments

Aim : To develop a java program to perform matrix addition.

Program :

```
import java.util.*;

class MatAdd {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter r value : ");

        int r = sc.nextInt();

        System.out.println("Enter c value : ");

        int c = sc.nextInt(), i, j;

        int a[][] = new int[r][c];

        int b[][] = new int[r][c];

        System.out.println("Enter matrix A elements : ");

        for (i = 0; i < r; i++) {

            for (j = 0; j < c; j++)

                a[i][j] = sc.nextInt();

        }

        System.out.println("Enter matrix B elements : ");

        for (i = 0; i < r; i++) {

            for (j = 0; j < c; j++)

                b[i][j] = sc.nextInt();

        }

        System.out.println("Addition : ");

        for (i = 0; i < r; i++) {

            for (j = 0; j < c; j++)

                System.out.print(a[i][j] + b[i][j] + " ");

            System.out.println();

        }

    }

}
```


output:

Enter r value :

3

Enter c value :

3

Enter matrix A elements :

1 2 3 4 5 6 7 8 9

Enter matrix B elements :

9 8 7 6 5 4 3 2 1

Addition :

10 10 10

10 10 10

10 10 10

Aim : To develop a java program to perform matrix multiplication.

Program :

```
import java.util.*;

class MatrixMul {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter r1 : ");

        int r1 = sc.nextInt();

        System.out.println("Enter c1 : ");

        int c1 = sc.nextInt();

        int i, j, k, c;

        int a[][] = new int[r1][c1];

        int b[][] = new int[r1][c1];

        System.out.println("Enter matrix A elements : ");

        for (i = 0; i < r1; i++) {

            for (j = 0; j < c1; j++)

                a[i][j] = sc.nextInt();
```

```

    }

    System.out.println("Enter matrix B elements : ");

    for (i = 0; i < r1; i++) {
        for (j = 0; j < c1; j++)
            b[i][j] = sc.nextInt();
    }

    System.out.println("Multiplication : ");

    for (i = 0; i < r1; i++) {
        for (j = 0; j < c1; j++) {
            c = 0;
            for (k = 0; k < c1; k++)
                c += a[i][k] * b[k][j];
            System.out.print(c + " ");
        }
        System.out.println();
    }
}

```

Output:

```

Enter r1 :
3
Enter c1 :
3
Enter matrix A elements :
1 2 3 4 5 6 7 8 9
Enter matrix B elements :
9 8 7 6 5 4 3 2 1
Multiplication :
30 24 18
84 69 54
138 114 90

```

Aim : To develop a java program to illustrate jagged array.

Program :

```
import java.util.*;

class JagArray {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter no of rows : ");

        int m = sc.nextInt();

        int n, i, j;

        int a[][] = new int[m][];

        for (i = 0; i < m; i++) {

            System.out.println("Enter no of columns of row " + (i + 1));

            n = sc.nextInt();

            a[i] = new int[n];

        }

        System.out.println("Enter jagged array elements : ");

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

            for (j = 0; j < a[i].length; j++)

                a[i][j] = sc.nextInt();

        }

        System.out.println("Jagged array elements are : ");

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

            for (j = 0; j < a[i].length; j++)

                System.out.print(a[i][j] + " ");

            System.out.println();

        }

    }

}
```

Output:

Enter no of rows :

3

Enter no of columns of row 1

2

Enter no of columns of row 2

3

Enter no of columns of row 3

4

Enter jagged array elements :

1 2

3 4 5

6 7 8 9

Jagged array elements are :

1 2

3 4 5

6 7 8 9

b)Construct a java program to illustrate non-static nested classes

Aim : To develop a java program to illustrate Member inner class.

Program:

```
class OuterClass {  
    int a = 10;  
    class InnerClass {  
        void disp() {  
            System.out.println(a);  
            System.out.println("Member Inner Class");  
        }  
    }  
}  
  
class MemberMain {  
    public static void main(String[] args) {  
        OuterClass o = new OuterClass();  
        OuterClass.InnerClass i = o.new InnerClass();  
        i.disp();  
    }  
}
```

```
}
```

Output:

10

Member Inner Class

Aim : To develop a java program to illustrate Local inner class.

Program :

```
class LocalOuter {  
    int a = 6264;  
    void print() {  
        class LocalInner {  
            void disp() {  
                System.out.println(a);  
                System.out.println("Local Inner Class");  
            }  
        }  
        LocalInner i = new LocalInner();  
        i.disp();  
    }  
}  
  
class LocalMain {  
    public static void main(String[] args) {  
        LocalOuter o = new LocalOuter();  
        o.print();  
    }  
}
```

Output:

6264

Local Inner Class

Aim : To develop a java program to illustrate StaticNested class.

Program :

```
class Outter {
```

```

static int a = 6264;
static class Inner {
    void disp() {
        System.out.println(a);
        System.out.println("Static Nested Class");
    }
}
}
class StaticNestedMain {
    public static void main(String[] args) {
        Outer.Inner o = new Outer.Inner();
        o.disp();
    }
}

```

Output:

6264

Static Nested Class

Aim : To develop a java program to perform addition of n number using command line arguments.

Program :

```

class CommandLineArgsUsingLoop
{
    public static void main(String args[])
    {
        for(int i=0;i<args.length;i++)
            System.out.println("Argument "+(i+1)+":"+args[i]);
    }
}

```

Output:

java CommandLineArgsUsingLoop 11 12 13

Argument 1:11

Argument 2:12

Argument 3:13

5. Construct programs using the following concepts.

a) **Inheritance** b) **Usage of super** c) **Method Overriding**

Aim : To develop a java program to illustrate Single Inheritance

Program :

```
class Emp{
    int sal;
}
class Faculty extends Emp {
    int al;
    Faculty(int a, int b) {
        sal = a;
        al = b;
    }
    void disp() {
        System.out.println("Sallary : " + sal+ " allowance : " + al);
    }
}
class SingleInheritance {
    public static void main(String[] args) {
        Faculty f = new Faculty(10000, 5000);
        f.disp();
    }
}
```

Output:

Sallary : 10000 allowance : 5000

Aim : To develop a java program to illustrate Multi Level Inheritance

Program :

```
class Animal {
    void eat() {
        System.out.println("Eating");
    }
}
```

```

}
class Dog extends Animal {
    void bark() {
        System.out.println("Barking");
    }
}
class Puppy extends Dog {
    void sleep() {
        System.out.println("Sleeping");
    }
}
class MultiLevelInheritance {
    public static void main(String[] args) {
        Puppy p = new Puppy();
        p.eat();
        p.bark();
        p.sleep();
    }
}

```

Output:

Eating

Barking

Sleeping

Aim : To develop a java program to demonstrate usage of super.

Program :

```

class A {
    int s = 10;

    A() {
        System.out.println("Parent Constructor");
    }

    void print() {
        System.out.println(s);
    }
}

class B extends A {

```



```

    int s = 20;

    B() {
        super();
    }

    void print() {
        System.out.println("Parent Class Value :");
        super.print();
        System.out.println("Child Class Value :");
        System.out.println(s);
    }
}

class SuperDemo {
    public static void main(String[] args) {
        B b = new B();
        b.print();
    }
}

```

Output:

```

Parent Constructor
Parent Class Value :
10
Child Class Value :
20

```

Aim : To develop a java program to illustrate Method Overloading.

Program :

```

class A {
    void print() {
        System.out.println("Hello");
    }
}

```

```
class B extends A {  
    void print() {  
        System.out.println("World");  
    }  
}  
  
class C extends B {  
    void print() {  
        System.out.println("SVEC");  
    }  
}  
  
class MethodOverlaoding {  
    public static void main(String[] args) {  
        C c = new C();  
        c.print();  
        A a = new A();  
        a.print();  
        B b=new B();  
        b.print();  
  
    }  
}
```

Output:

SVEC

Hello

World

6. Construct programs using the following concepts.

a) Usage of final b) Abstract class c) Interfaces

Aim : To develop a java program to demonstrate usage of final.

Program :

```
class FinalDemo {
    final int b;
    FinalDemo(int b) {
        this.b = b;
    }
    final void disp() {
        System.out.println(b);
    }
    void update(int s) {
        b = s; // final variable can't change once initialize - error
    }
}

class FinalSample extends FinalDemo {
    final void disp() {
        System.out.println(b); // final variable cannot overridden - error
    }
}

class FinalMain {
    public static void main(String[] args) {
        FinalDemo f = new FinalDemo(2552);
        f.disp();
        f.update(20);
        FinalSample f1 = new FinalSample();
        f1.disp();

    }
}
```

Output :

10

Exception in thread "main" java.lang.Error: Unresolved compilation problem:
The final field FinalDemo.b cannot be assigned

at FinalDemo.update(FinalMain.java:13)
at FinalMain.main(FinalMain.java:27)

Aim : To develop a java program to demonstrate abstract class.

Program :

```
abstract class Shape {  
    abstract void draw();  
    void disp() {  
        System.out.println("Shape");  
    }  
}  
  
class Rect extends Shape {  
    void draw() {  
        System.out.println("Rectangle");  
    }  
}  
  
class Circ extends Shape {  
    void draw() {  
        System.out.println("Circle");  
    }  
}  
  
class AbstractMain {  
    public static void main(String[] args) {  
        Rect r = new Rect();  
        Circ c = new Circ();  
        r.draw();  
    }  
}
```

```

        r.disp();
        c.draw();
        c.disp();
    }
}

```

Output:

Rectangle

Shape

Circle

Shape

Aim : To develop a java program to demonstrate interfaces.

Program :

```

interface Sample {
    int a = 2662;
    void disp();
}

interface Demo {
    void print();
}

class Sam implements Sample, Demo {
    public void disp() {
        System.out.println("Sample");
    }
    public void print() {
        System.out.println("Demo");
    }
    void sam() {
        System.out.println("sam");
    }
}

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

```

```
Sam s = new Sam();  
s.disp();  
s.print();  
s.sam();  
System.out.println(s.a);  
}  
}
```

Outut:

Sample

Demo

sam

2662

7. Implement the programs using the concepts

a) Packages b) Exception Handling

Aim : To develop a java program to demonstrate built in packages.

Program :

```
import java.util.Scanner; // importing built package util.Scanner
```

```
class BuiltInPack{
```

```
    public static void main(String args[]) {
```

```
        Scanner s = new Scanner(System.in);
```

```
        System.out.println("Enter number: ");
```

```
        int n = s.nextInt();
```

```
        System.out.println("Given number: ");
```

```
        System.out.println(n);
```

```
        System.out.println("Enter float: ");
```

```
        float f = s.nextFloat();
```

```
        System.out.println("Given float: ");
```

```
        System.out.println(f);
```

```
        System.out.println("Enter string: ");
```

```
        s.nextLine();
```

```
        String st = s.nextLine();
```

```
        System.out.println("Given string: " + st);
```

```
    }
```

```
}
```

Output:

Enter number:

6264

Given number:

6264

Enter float:

82.76

Given float:

82.76

Enter string:

1@3\$5^7*9)

Given string: 1@3\$5^7*9)

Aim : To develop a java program to demonstrate user defined packages.

Program :

```
package pack;
public class SamplePackage {
    public int a;

    public SamplePackage(int a) {
        this.a = a;
    }
    public void disp() {
        System.out.println(a);
    }
}

import pack.SamplePackage;
class PackageDemo {
    public static void main(String[] args) {
        SamplePackage s = new SamplePackage(20);
        System.out.println(s.a);
        s.disp();
    }
}
```

Output:

E:\java_lab>javac -d . SamplePackage.java

E:\java_lab>javac PackageDemo.java

E:\java_lab>java PackageDemo

20

20

Aim : To develop a java program to demonstrate built in Exception.

Program :

```
class TestBuiltInException {  
    public static void main(String[] args) {  
        int a;  
        try {  
            a = 50 / 0;  
        } catch (ArithmeticException e) {  
            System.out.println(e);  
        } catch (Exception e) {  
            System.out.println(e);  
        } finally {  
            System.out.println("ExceptionHandling");  
        }  
        System.out.println("rest of code");  
    }  
}
```

Output:

```
java.lang.ArithmeticException: / by zero  
ExceptionHandling  
rest of code
```

Aim : To develop a java program to demonstrate built in Exception.

Program :

```
class UserDefinedExcept extends Exception {  
    UserDefinedExcept(String s) {  
        super(s);  
    }  
}  
  
class TestUserExcept {  
    static void validate() throws UserDefinedExcept {  
        throw new UserDefinedExcept("not eligible to vote");  
    }  
    public static void main(String[] args) {  
        try {  
            validate();  
        } catch (UserDefinedExcept e) {  
            e.printStackTrace();  
        }  
        System.out.println("rest of code");  
    }  
}
```

Output:

```
UserDefinedExcept: not eligible to vote  
    at TestUserExcept.validate(TestUserExcept.java:8)  
    at TestUserExcept.main(TestUserExcept.java:12)  
rest of code
```

8.Implement the programs on Multi-Threading.

a)Multiple Threads on Single Object b) Thread Deadlock

Aim : To develop a java program to demonstrate Multiple threads on single object without synchronized.

Program :

```
class Table {  
    void printTab(int n) {  
        for (int i = 1; i <= 5; i++) {  
            try {  
                Thread.sleep(500);  
            } catch (Exception e) {  
                System.out.println(e);  
            }  
            System.out.println(i + "x" + n + "=" + (i * n));  
        }  
    }  
}  
  
class Th1 extends Thread {  
    Table t;  
    Th1(Table t) {  
        this.t = t;  
    }  
    public void run() {  
        t.printTab(5);  
    }  
}  
  
class Th2 extends Thread {  
    Table t;  
    Th2(Table t) {  
        this.t = t;  
    }  
}
```

```

        public void run() {
            t.printTab(10);
        }
    }
}

class MultipleThreads {
    public static void main(String[] args){
        Table t = new Table();
        Th1 t1 = new Th1(t);
        Th2 t2 = new Th2(t);
        t1.start();
        t2.start();
    }
}

```

Output:

```

1x5=5
1x10=10
2x10=20
2x5=10
3x10=30
3x5=15
4x5=20
4x10=40
5x10=50
5x5=25

```

Aim : To develop a java program to demonstrate Multiple threads on single object with synchronized.

Program :

```

class Table {
    synchronized void printTab(int n) {
        for (int i = 1; i <= 5; i++) {
            try {

```

```

        Thread.sleep(500);
    } catch (Exception e) {
        System.out.println(e);
    }
    System.out.println(i + "x" + n + "=" + (i * n));
}
}
}

class Th1 extends Thread {
    Table t;

    Th1(Table t) {
        this.t = t;
    }

    public void run() {
        t.printTab(5);
    }
}

class Th2 extends Thread {
    Table t;

    Th2(Table t) {
        this.t = t;
    }

    public void run() {
        t.printTab(10);
    }
}

class MultipleThreadsDemo {
    public static void main(String[] args){
        Table t = new Table();
        Th1 t1 = new Th1(t);
        Th2 t2 = new Th2(t);
        t1.start();

```

```
        t2.start();
    }
}
```

Output:

```
1x5=5
2x5=10
3x5=15
4x5=20
5x5=25
1x10=10
2x10=20
3x10=30
4x10=40
5x10=50
```

Aim : To develop a java program to demonstrate thread deadlock.

Program :

```
class Deadlock {
    public static void main(String[] args) {
        final String r1 = "Printer";
        final String r2 = "Scanner";
        Thread t1 = new Thread() {
            public void run() {
                synchronized (r1) {
                    System.out.println("Thread 1: Locked " + r1);
                    synchronized (r2) {
                        System.out.println("Thread 1: Locked " + r2);
                    }
                }
            }
        };
        Thread t2 = new Thread() {
            public void run() {
```

```
synchronized (r2) {  
    System.out.println("Thread 2: Locked " + r2);  
    synchronized (r1) {  
        System.out.println("Thread 1: Locked " + r1);  
    }  
}  
};  
t1.start();  
t2.start();  
}
```

Output:

Thread 1: Locked Printer

Thread 2: Locked Scanner

9. Construct a program that shows Inter-thread Communication

Aim : To develop a java program to demonstrate Multiple threads on single object without synchronized.

Program :

```
class Customer {  
    int amount = 10000;  
  
    synchronized void withdraw(int amount) {  
        if (this.amount < amount) {  
            System.out.println("Less Balance");  
            try {  
                wait();  
            } catch (Exception e) {  
                System.out.println(e);  
            }  
        }  
        this.amount -= amount;  
        System.out.println("Amount has been Withdrawn");  
    }  
  
    synchronized void deposit(int amount) {  
        this.amount += amount;  
        System.out.println("Amount is Deposited");  
        notify();  
    }  
}  
  
class T1 extends Thread {  
    Customer c;  
  
    T1(Customer c) {  
        this.c = c;  
    }  
  
    public void run() {
```



```

        c.withdraw(15000);
    }
}

class T2 extends Thread {
    Customer c;
    T2(Customer c) {
        this.c = c;
    }
    public void run() {
        c.deposit(10000);
    }
}

class InterThreadCom{
    public static void main(String args[]) {
        Customer c = new Customer();
        T1 t1 = new T1(c);
        T2 t2 = new T2(c);
        t1.start();
        t2.start();
    }
}

```

Output:

Less Balance

Amount is Deposited

Amount has been Withdrawn

10. Construct programs to perform read and write operations on files.

a) Sequential Files

b) Random Access files

Aim : To develop a java program to retrieve data from file.

Program :

```
import java.io.*;
class FileInputStreamExample {
    public static void main(String[] args) {
        try {
            FileInputStream f = new FileInputStream("sam.txt");
            int i;
            while ((i = f.read()) != -1)
                System.out.print((char) i);
            f.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

Output:

Hello World!

I'm chatBot

Aim : To develop a java program to write data into file.

Program :

```
import java.io.*;
import java.util.*;
class FileOutputStreamEx {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            String s, m = "";
            FileOutputStream f = new FileOutputStream("sam.txt");
            System.out.println("write the content :");
            while (!(s = sc.nextLine()).equals("bye"))
                m += s;
        }
    }
}
```

```

        f.write(m.getBytes());
        f.close();
    } catch (Exception e) {
        System.out.println(e);
    } finally {
        System.out.print("work is done");
    }
}
}

```

Output:

write the content :

hello

cai IV Sem

bye

work is done

Aim : To develop a java program to copy data from one file to another file.

Program :

```

import java.io.*;

class CopyFile {

    public static void main(String[] args) {

        try {

            FileWriter fw = new FileWriter("sam1.txt");
            FileReader fr = new FileReader("sam.txt");

            int i;

            String s = "";

            while ((i = fr.read()) != -1)

                s += (char)i;

            fw.write(s);

            fw.close();

            fr.close();

        } catch (Exception e) {

```

```

        System.out.println(e);
    } finally {
        System.out.print("work is done");
    }
}
}
}

```

Output :

work is done

Aim : To develop a java program to demonstrate random access file.

Program :

```

import java.io.*;
import java.util.Scanner;

public class RandomAccessFileEx {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        String filePath = "sam.txt", s;
        try {

            RandomAccessFile f = new RandomAccessFile(filePath, "rw");
            f.seek(f.length());
            System.out.println("write the content :");
            s = sc.nextLine();
            f.write(s.getBytes()); // write into file
            f.seek(0);
            System.out.println("File content: ");
            while ((s = f.readLine()) != null) {

                System.out.println(s); // getting data from file
            }
            f.close();
        } catch (IOException e) {

            System.out.println(e);
        }
    }
}

```

```
}  
}  
}
```

Output:

write the content :

hello

File content:

hello cai IV Semhello

11. Construct a program to demonstrate ArrayList .

Aim : To develop a menu driven java program to demonstrate ArrayList operations.

Program :

```
import java.util.*;

class ArrayListDemo {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        ArrayList<Integer> al = new ArrayList<Integer>();

        int c, key, l;

        while (true) {

            System.out.println("choose options :\n 1)insert \n 2)update \n 3)search \n 4)display \n 5)exit");

            c = sc.nextInt();

            l = al.size();

            switch (c) {

                case 1: {

                    System.out.println("Enter the element :");

                    al.add(sc.nextInt());

                }

                break;

                case 2: {

                    if (l == 0)

                        System.out.println("No elements in list to update");

                    else {

                        System.out.println("Enter the element index to update :");

                        key = sc.nextInt();

                        if (key >= l)

                            System.out.println("index is out of the range ");

                        else {

                            System.out.println("Enter the element :");
```

```

        al.set(key, sc.nextInt());
    }
}
break;
case 3: {
    if (l == 0)
        System.out.println("No elements in list");
    else {
        System.out.println("Enter the key element :");
        key = sc.nextInt();
        if (al.contains(key))
            System.out.println("Element is found at index :" + al.indexOf(key));
        else
            System.out.println("Element is not found");
    }
}
break;
case 4: {
    if (l == 0)
        System.out.println("No elements in list");
    else {
        System.out.println("Element are :");
        for (Integer e : al) {
            System.out.print(e + " ");
        }
        System.out.println();
    }
}
break;
case 5:
    System.exit(0);

```

```

        default:
            System.out.println("choose the correct option .....");
        }
        System.out.println(".....");
    }
}
}

```

Output:

choose options :

- 1)insert
- 2)update
- 3)search
- 4)display
- 5)exit

1

Enter the element :

10

.....

choose options :

- 1)insert
- 2)update
- 3)search
- 4)display
- 5)exit

1

Enter the element :

20

.....

choose options :

- 1)insert
- 2)update
- 3)search

4)display

5)exit

2

Enter the element index to update :

1

Enter the element :

50

.....

choose options :

1)insert

2)update

3)search

4)display

5)exit

3

Enter the key element :

50

Element is found at index :1

.....

choose options :

1)insert

2)update

3)search

4)display

5)exit

4

Element are :

10 50

.....

12. Construct a program to demonstrate LinkedList .

Aim : To develop a java program to hold N products objects in LinkedList.

Program :

```
import java.util.*;

class Product {

    String pid;

    float r;

    Product(String pid, float r) {

        this.pid = pid;

        this.r = r;

    }

}

class LinkedListDemo1111 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        LinkedList<Product> plist = new LinkedList<Product>();

        int n, i;

        String pid;

        float r;

        System.out.println("Enter the no of products :");

        n = sc.nextInt();

        for (i = 0; i < n; i++) {

            System.out.println("Enter the product id :");

            pid = sc.next();

            System.out.println("Enter the product rate :");

            r = sc.nextFloat();

            plist.add(new Product(pid, r));

        }

        System.out.println("| Product id | rate |");

        System.out.println("-----");

        for (Product p : plist)

            System.out.println("| " + p.pid + " | " + p.r + " |");

    }

}
```

```
        System.out.println("-----");
    }
}
```

Output:

Enter the no of products :

3

Enter the product id :

12a1

Enter the product rate :

450

Enter the product id :

12a2

Enter the product rate :

300

Enter the product id :

12a3

Enter the product rate :

999

| Product id | rate |

| 12a1 | 450.0 |

| 12a2 | 300.0 |

| 12a3 | 999.0 |

Aim : To develop a java program to illustrate polynomial using LinkedList.

Program :

```
import java.util.*;

class Polynomial {
    int e, c;

    Polynomial(int c, int e) {
        this.e = e;
        this.c = c;
    }
}

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

        LinkedList<Polynomial> poly = new LinkedList<Polynomial>();

        int n, i, c;
        String s = "";

        System.out.println("Enter the degree of polynomial:");
        n = sc.nextInt();

        for (i = 0; i <= n; i++) {
            System.out.println("Enter the coefficient of x^" + (n - i) + " :");
            c = sc.nextInt();
            poly.add(new Polynomial(c, n - i));
        }

        System.out.println("Polynomial :");
        for (Polynomial p : poly){
            if (p.c == 0)
                continue;

            s += Integer.toString(p.c) + "x^" + Integer.toString(p.e) + " + ";
        }
        s = s.substring(0, s.length() - 6);

        System.out.println(s);
    }
}
```

```
}
```

Output:

Enter the degree of polynominal:

3

Enter the coefficent of x^3 :

4

Enter the coefficent of x^2 :

3

Enter the coefficent of x^1 :

2

Enter the coefficent of x^0 :

1

Polynomial :

$4x^3 + 3x^2 + 2x^1 + 1$

Aim: Construct a java program to iterate elements of HashSet using iterator.

Program:

```
import java.util.*;

class HashSetIteration {

    public static void main(String[] args) {

        HashSet<Integer> s = new HashSet<>();

        s.add(10);
        s.add(20);
        s.add(30);

        System.out.println("Iterating HashSet using Iterator:");

        Iterator<Integer> i = s.iterator();

        while (i.hasNext()) {

            System.out.println(i.next());

        }

    }

}
```

Output:

Iterating HashSet using Iterator:

20

10

30

AIM: Construct a java program to iterate elements of LinkedHashSet using for Each

PROGRAM:

```
import java.util.*;

class LinkedHashSetIteration {

    public static void main(String[] args) {

        LinkedHashSet<Integer> s = new LinkedHashSet<>();

        s.add(10);

        s.add(20);

        s.add(30);

        System.out.println("Using for-each loop:");

        for (Integer e : s) {

            System.out.println(e);

        }

    }

}
```

Output:

Using for-each loop:

10

20

30

AIM: Construct a java program for TreeSet

PROGRAM:

```
import java.util.*;

class Product implements Comparable<Product> {
```

```

    int id;

    String name;

    Product(int id, String name) {
        this.id = id;
        this.name = name;
    }

    public int compareTo(Product p) {
        if (id < p.id)
            return -1;
        else if (id > p.id)
            return 1;
        else
            return 0;
    }
}

class TreeSetDemo {
    public static void main(String args[]) {
        TreeSet<Product> t = new TreeSet<Product>();
        t.add(new Product(20, "pen"));
        t.add(new Product(10, "pencil"));
        Iterator<Product> i = t.iterator();
        while (i.hasNext()) {
            Product u = i.next();
            System.out.println(u.id + " " + u.name);
        }
    }
}

```

Output:

10 pencil

20 pen

Aim: Construct a java program for HashMap.

Program:

```
import java.util.*;

class Product {

    int id;

    String name;

    Product(int id, String name) {

        this.id = id;

        this.name = name;

    }

}

class HashMapDemo {

    public static void main(String aergs[]) {

        HashMap<Integer, Product> h = new HashMap<Integer, Product>();

        h.put(1, new Product(101, "Pen"));

        h.put(2, new Product(102, "Pencil"));

        h.put(3, new Product(103, "Book"));

        for (Map.Entry<Integer, Product> i : h.entrySet()) {

            Product s = i.getValue();

            System.out.println(i.getKey());

            System.out.println(s.id + " " + s.name);

        }

    }

}
```

Output:

```
1
101 Pen
2
102 Pencil
3
103 Book
```