**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 StudentGrade {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("enter the marks of student : ");

int n = sc.nextInt();

if (n > 90)

System.out.println("A+ Grade");

else if (n > 80)

System.out.println("A Grade");

else if (n > 70)

System.out.println("B Grade");

else if (n > 60)

System.out.println("C Grade");

else if (n > 50)

System.out.println("D Grade");

else if (n >= 35)

System.out.println("E Grade");

else

System.out.println("FAIL");

}

}

**Output :**

enter the marks of student :

45

E 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);

int n, f = 1;

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

n = sc.nextInt();

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

f \*= i;

}

System.out.println("factorial of given number is " + f);

}

}

**Output :**

enter n value :

5

The factorial of 5 is 120

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

**Program :**

import java.util.\*;

class Prime {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt(), i = 2;

while (i <= n / 2) {

if ((n % i) == 0) {

System.out.println(n + " is Not a Prime");

break;

}

i++;

}

if (i > n / 2)

System.out.println(n + " is a Prime");

}

}

**Output :**

enter n value :

2

2 is a Prime

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

**Program :**

import java.util.\*;

class IntegerPalindrome {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("enter number : ");

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

while (n > 0) {

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

n /= 10;

}

if (t == a)

System.out.println(t + " is a palindrome");

else

System.out.println(t + " is Not a palindrome");

}

}

**Output 1:**

enter number :

12321

12321 is a palindrome

**Output 2:**

enter number :

456321

456321 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 :

5

enter n value :

25

Prime numbers are :

5 7 11 13 17 19 23

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

**Program :**

class TypeCastCon {

public static void main(String args[]) {

int x = 30;

float f = x; // Type conversion

System.out.println(x + " integer");

System.out.println(f + " float");

float m = 10.0f;

int n = (int) m; // Type casting

System.out.println(m + " float");

System.out.println(n + " integer");

// integer to char

char c = (char) x;

System.out.println(c);

}

}

**Output :**

30 integer

30.0 float

10.0 float

10 integer

▲

**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 i = 0;

int n = s.length() - 1;

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 :

Malayalam

Palindrome

**Output 2:**

enter the String :

telugu

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 {

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, ic = 0, oc = 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')

ic++;

else

spc++;

if (conVowel(c))

oc++;

else if (c >= 'A' && c <= 'z')

cc++;

else

;

}

System.out.println("lowercase : " + lc);

System.out.println("uppercase : " + uc);

System.out.println("integers : " + ic);

System.out.println("vowels : " + oc);

System.out.println("consonants : " + cc);

System.out.println("special characters : " + spc);

}

static boolean conVowel(char c) {

int i;

char a[] = { 'a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U' };

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

if (c == a[i])

return true;

}

return false;

}

}

**Output:**

AS@DFTYU2+=567bvcAER9@Tbhi55o1p

lowercase : 8

uppercase : 11

integers : 8

vowels : 6

consonants : 13

special characters : 4

**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 StdMain {

public static void main(String args[]) {

Std s = new Std();

s.read();

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

s.disp();

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

}

**Output :**

enter Student id :

21A81A

enter Student name :

SURYA

enter Student cgpa :

9.3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

rno : 21A81A

name : SURYA

cgpa : 9.3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**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);

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 StdArrayMain {

public static void main(String args[]) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

int i;

StdArray s[] = new StdArray[n];

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

s[i] = new StdArray();

s[i].read();

}

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

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

s[i].disp();

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

}

}

**Output :**

enter no of Students :

3

enter Student id :

21A81A1

enter Student name :

SURYA

enter Student cgpa :

9.3

enter Student id :

21A81A2

enter Student name :

RADHA

enter Student cgpa :

9.4

enter Student id :

21A81A3

enter Student name :

SURESH

enter Student cgpa :

9.1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

rno : 21A81A1

name : SURYA

cgpa : 9.3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

rno : 21A81A2

name : RADHA

cgpa : 9.4

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

rno : 21A81A3

name : SURESH

cgpa : 9.1

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

**Program :**

class Demo {

static int a = 10;

static int b = 20;

static void disp() {

System.out.println(b);

System.out.println("static method disp ");

}

}

class StaticDemo {

public static void main(String args[]) {

System.out.println(Demo.a);

Demo.disp();

}

}

**Output :**

10

20

static method disp

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

**Program :**

class DefaultAndParameter {

String s;

DefaultAndParameter() {

System.out.println("default constructor");

}

DefaultAndParameter(String x) {

s = x;

System.out.println(s);

System.out.println("parameter constructor");

}

}

class DefaultAndParameterConstructor {

public static void main(String[] args) {

DefaultAndParameter d = new DefaultAndParameter();

DefaultAndParameter p = new DefaultAndParameter("Hello ,World!");

}

}

**Output :**

default constructor

Hello ,World!

parameter constructor

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

**Program :**

class Sample1{

String s;

int i;

Sample1() {

System.out.println("Default constructor");

}

Sample1(String x) {

s = x;

System.out.println("String constructor");

}

Sample1(int x) {

i = x;

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

}

}

class ConstructoOverloading {

public static void main(String[] args) {

Sample1 s1 = new Sample1();

Sample1 s2 = new Sample1(10);

Sample1 s3 = new Sample1("suresh");

}

}

**Output :**

Default constructor

Integer constructor

String constructor

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

**Program :**

class SampleCopy {

String s;

int i;

SampleCopy() {

s = "Suresh";

i = 49;

SampleCopy(SampleCopy x) {

this.s = x.s;

this.i = x.i;

}

}

class CopyConstructor {

public static void main(String[] args) {

SampleCopy s1 = new SampleCopy();

SampleCopy s2;

s2 = new SampleCopy(s1);

System.out.println(" S1 values : " + s1.s + " " + s1.i);

System.out.println(" S2 values : " + s2.s + " " + s2.i);

}

}

**Output :**

S1 values : Suresh 49

S2 values : Suresh 49

**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 MatrixAdd {

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, j;

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

int b[][] = new int[m][n];

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

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

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

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

}

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

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

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

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

}

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

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

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

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

System.out.println();

}

}

}

**Output :**

enter m value :

3

enter n 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 1 2 3 1 2 3

Enter matrix B elements :

1 2 3 1 2 3 1 2 3

Multiplication :

6 12 18

6 12 18

6 12 18

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

**Program :**

import java.util.\*;

class JaggedArray {

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 colums 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 :

2

enter no of colums of row 1

4

enter no of colums of row 2

3

Enter jagged array elements :

1 2 3 4 5 6 7

jagged array elements are :

1 2 3 4

5 6 7

**Construct a java program to illustrate non-static nested classes**

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

**Program :**

class Moutter {

int a = 10;

class Minner {

void disp() {

System.out.println(a);

System.out.println("Member Inner Class");

}

}

}

class MemberMain {

public static void main(String[] args) {

Moutter o = new Moutter();

Moutter.Minner i = o.new Minner();

i.disp();

}

}

**Output :**

10

Member Inner Class

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

**Program :**

class Loutter {

int a = 100;

void print() {

class Linner {

void disp() {

System.out.println(a);

System.out.println("Local Inner Class");

}

}

Linner i = new Linner();

i.disp();

}

}

class LocalMain {

public static void main(String[] args) {

Loutter o = new Loutter();

o.print();

}

}

**Output :**

100

Local Inner Class

**Aim :** To develop a java program to illustrate Single Inheritance

**Program :**

class Outter {

static int a = 10;;

static class Inner {

void disp() {

System.out.println(a);

System.out.println("Static Nested Class");

}

}

}

class StaticNestedMain {

public static void main(String[] args) {

Outter.Inner o = new Outter.Inner();

o.disp();

}

}

**Output :**

10

Static Nested Class

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

**Program :**

class SumOfNumbers {

public static void main(String a[]) {

int s = 0;

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

s = s + Integer.parseInt(a[i]);

System.out.print("sum of given numbers is : " + s);

}

}

**Output :**

D:\sem-4\javalab>java SumOfNumbers 10 20 30 40

sum of given numbers is : 100

**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 Employee {

int salary;

}

class Faculty extends Employee {

int allowance;

Faculty(int a, int b) {

salary = a;

allowance = b;

}

void disp() {

System.out.println("Sallary : " + salary + " allowance : " + allowance);

}

}

class SingleInheritance {

public static void main(String[] args) {

Faculty f = new Faculty(100, 50);

f.disp();

}

}

**Output :**

Sallary : 100 allowance : 50

**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 = 20;

A() {

System.out.println("parent constructor");

}

void print() {

System.out.println("Hello");

System.out.println(s);

}

}

class B extends A {

int s = 30;

B() {

super();

}

void print() {

System.out.println("parent class values :");

super.print();

System.out.println("World");

System.out.println(s);

}

}

class SuperDemo {

public static void main(String[] args) {

B b = new B();

b.print();

}

}

**Output :**

parent constructor

parent class values :

Hello

20

World

30

**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();

}

}

**Output :**

SVEC

Hello

**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 intiallize - 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(10);

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 = 20;

void disp();

}

interface Demo {

void prit();

}

class Sam implements Sample, Demo {

public void disp() {

System.out.println("Sample");

}

public void prit() {

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.prit();

s.sam();

System.out.println(s.a);

}

}

**Output :**

Sample

Demo

sam

20

**7. Implement the programs using the concepts**

**a)Packages b) Exception Handling**

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

**Program :**

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

class BuiltInPackagesDemo {

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:

15

Given number:

15

Enter float:

2.3

Given float:

2.3

Enter string:

@1a81A4349

Given string: @1a81A4349

**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 :**

D:\sem-4\javalab\java programing lab> javac -d . SamplePackage.java

D:\sem-4\javalab\java programing lab> javac PackageDemo.java

D:\sem-4\javalab\java programing lab> java PackageDemo

20

20

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

**Program :**

class TestBuiltInException {

public static void main(String[] args) {

int a[] = new int[3];

try {

a[4] = 50 / 0;

} catch (ArithmeticException e) {

System.out.println(e);

} catch (ArrayIndexOutOfBoundsException 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 biuilt in Exception.

**Program :**

class UserDefinedException extends Exception {

UserDefinedException(String s) {

super(s);

}

}

class TestUserException {

static void validate() throws UserDefinedException {

throw new UserDefinedException("not eligible to vote");

}

public static void main(String[] args) {

try {

validate();

} catch (UserDefinedException e) {

e.printStackTrace();

}

System.out.println("rest of code");

}

}

**Output :**

UserDefinedException: not eligible to vote

at TestUserException.validate(TestUserException.java:9)

at TestUserException.main(TestUserException.java:14)

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(100);

}

}

class Th2 extends Thread {

Table t;

Th2(Table t) {

this.t = t;

}

public void run() {

t.printTab(500);

}

}

class MultipleThreadsSingleObject {

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 :**

1x500=500

1x100=100

2x500=1000

2x100=200

3x100=300

3x500=1500

4x100=400

4x500=2000

5x100=500

5x500=2500

**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(100);

}

}

class Th2 extends Thread {

Table t;

Th2(Table t) {

this.t = t;

}

public void run() {

t.printTab(500);

}

}

class MultipleThreadsSingleObject {

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 :**

1x100=100

2x100=200

3x100=300

4x100=400

5x100=500

1x500=500

2x500=1000

3x500=1500

4x500=2000

5x500=2500

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

**Program :**

class DeadLock {

synchronized void printer(DeadLock d2) {

System.out.println("printer start");

try {

Thread.sleep(2000);

} catch (Exception e) {

System.out.println(e);

}

d2.scanner(this);

System.out.println("printer end");

}

synchronized void scanner(DeadLock d1) {

try {

Thread.sleep(2000);

} catch (Exception e) {

System.out.println(e);

}

System.out.println("scanner start");

d1.printer(this);

System.out.println("scanner end");

}

}

class Thread1 extends Thread {

DeadLock d1, d2;

Thread1(DeadLock d1, DeadLock d2) {

this.d1 = d1;

this.d2 = d2;

}

public void run() {

d1.printer(d2);

}

}

class Thread2 extends Thread {

DeadLock d1, d2;

Thread2(DeadLock d1, DeadLock d2) {

this.d1 = d1;

this.d2 = d2;

}

public void run() {

d2.scanner(d1);

}

}

class ThreadDeadLockTest {

public static void main(String[] args) {

DeadLock d1 = new DeadLock();

DeadLock d2 = new DeadLock();

Thread1 t1 = new Thread1(d1, d2);

Thread2 t2 = new Thread2(d1, d2);

t1.start();

t2.start();

}

}

**Output :**

printer start

scanner start

**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("Low balance");

try {

wait();

} catch (Exception e) {

System.out.println(e);

}

}

this.amount -= amount;

System.out.println("amount has 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(1000);

}

}

class InterThreadCommunication {

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 :**

Low balance

amount is deposited

amount has 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 retrive 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 new version of AI

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

**Program :**

import java.io.\*;

import java.util.\*;

class FileOutputStreamExample {

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 :

desgining a new tool for java

this tool contains packages and certain class and interfaces

bye

work is done

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

**Program :**

import java.io.\*;

class CopyingFile {

public static void main(String[] args) {

try {

FileWriter fw = new FileWriter("sam1.txt");

FileReader fr = new FileReader("sam2.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 RandomAccessFileExample {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String filePath = "sam.txt", s;

try {

RandomAccessFile file = new RandomAccessFile(filePath, "rw");

file.seek(file.length());

System.out.println("write the content :");

s = sc.nextLine();

file.write(s.getBytes());// write into file

file.seek(0);

System.out.println("File content: ");

while ((s = file.readLine()) != null) {

System.out.println(s);// getting data from file

}

file.close();

} catch (IOException e) {

System.out.println(e);

}

}

}

**Output :**

write the content :

i'm new version of AI

File content:

hello,world!i'm new version of AI

**11.Construct a program to demonstrate ArrayList .**

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

**Program :**

import java.util.\*;

class ArrayListExample {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int o, s, i, l;

while (true) {

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

o = sc.nextInt();

l = al.size();

switch (o) {

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 :");

s = sc.nextInt();

if (s >= l)

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

else {

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

al.set(s, sc.nextInt());

}

}

} break;

case 3: {

if (l == 0)

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

else {

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

s = sc.nextInt();

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

if (al.get(i) == s) {

System.out.println("Element is found at index :" + i);

break;

}

}

if (i == l)

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 :

25

..............................

choose options :

1)insert

2)update

3)search

4)display

5)exit

2

Enter the element index to update :

3

index is out of the range

..............................

choose options :

1)insert

2)update

3)search

4)display

5)exit

1

Enter the element :

26

..............................

choose options :

1)insert

2)update

3)search

4)display

5)exit

2

Enter the element index to update :

1

Enter the element :

25

..............................

choose options :

1)insert

2)update

3)search

4)display

5)exit

4

Element are :

25 25

..............................

choose options :

1)insert

2)update

3)search

4)display

5)exit

3

Enter the key element :

25

Element is found at index :0

..............................

choose options :

1)insert

2)update

3)search

4)display

5)exit

5

**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 rating;

Product(String pid, float rating) {

for (int i = pid.length(); i < 10; i++)

pid = "0" + pid;

this.pid = pid;

this.rating = rating;

}

}

class LinkedListExample {

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 rating :");

r = sc.nextFloat();

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

}

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

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

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

for (Product p : plist)

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

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

}

}

**Output :**

Enter the no of products :

3

Enter the product id :

965823456

Enter the product rating :

3.5

Enter the product id :

12654

Enter the product rating :

4

Enter the product id :

98547632

Enter the product rating :

2.6

------------------------------

| Product id | rating |

-------------------------------

| 0965823456 | 3.5 |

| 0000012654 | 4.0 |

| 0098547632 | 2.6 |

-----------------------------

**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 polynominal:");

n = sc.nextInt();

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

System.out.println("Enter the coefficent of x^" + (n - i) + " :");

c = sc.nextInt();

poly.add(new Polynomial(c, n - i));

}

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

for (Polynomial p : poly)

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 :

5

Enter the coefficent of x^2 :

3

Enter the coefficent of x^1 :

8

Enter the coefficent of x^0 :

21

Polynomial :

5x^3 + 3x^2 + 8x^1 + 21