LAB PROGRAM 1A): Design and develop a Java program that reads two matrices A (m x n) and B (p x q) and Compute product of matrices A and B. Print both the input matrices and resultant matrix appropriately.

```
import java.util.Scanner;


public class Lab_program1a {


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


        System.out.print("Enter the number of rows (m) for matrix A: ");
        int m = scanner.nextInt();


        System.out.print("Enter the number of columns (n) for matrix A:
");
        int n = scanner.nextInt();


        System.out.print("Enter the number of rows (p) for matrix B: ");
        int p = scanner.nextInt();


        System.out.print("Enter the number of columns (q) for matrix B:
");
        int q = scanner.nextInt();


        if (n != p) {
            System.out.println("Error: The number of columns in matrix A
must be equal to the number of rows in matrix B.");
            return;
        }


        int[][] matrixA = new int[m][n];
        int[][] matrixB = new int[p][q];
```

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

        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                matrixA[i][j] = scanner.nextInt();
            }
        }

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

        for (int i = 0; i < p; i++) {
            for (int j = 0; j < q; j++) {
                matrixB[i][j] = scanner.nextInt();
            }
        }

        int[][] resultMatrix = multiplyMatrices(matrixA, matrixB);

        System.out.println("Matrix A:");
        printMatrix(matrixA);

        System.out.println("Matrix B:");
        printMatrix(matrixB);

        System.out.println("Resultant Matrix:");
        printMatrix(resultMatrix);
    }

    private static int[][] multiplyMatrices(int[][] matrixA, int[][]
matrixB) {
        int m = matrixA.length;
        int n = matrixA[0].length;
        int p = matrixB.length;
```

```
        int q = matrixB[0].length;

        int[][] result = new int[m][q];

        for (int i = 0; i < m; i++) {
            for (int j = 0; j < q; j++) {
                for (int k = 0; k < n; k++) {
                    result[i][j] += matrixA[i][k] * matrixB[k][j];
                }
            }
        }

        return result;
    }

    private static void printMatrix(int[][] matrix) {
        int rows = matrix.length;
        int columns = matrix[0].length;

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

LAB_PROGRAM 1B): Design and develop a Java Program to implement an Inventory Management System using Constructors.

```
import java.util.ArrayList;
class Product {
  String name;
  double price;
  int quantity;

  // Default Constructor
  public Product() {
      this.name = "Unnamed Product";
      this.price = 0.0;
      this.quantity = 0;
      System.out.println("Default Constructor called: " + name);
  }

  // Parameterized Constructor
  public Product(String name, double price, int quantity) {
      this.name = name;
      this.price = price;
      this.quantity = quantity;
      System.out.println("Parameterized Constructor called: " +
name);
  }

  // Copy Constructor
  public Product(Product product) {
      this.name = product.name;
      this.price = product.price;
```

```
        this.quantity = product.quantity;
        System.out.println("Copy Constructor called: " + name);
    }

    // Method to display product details
    public void displayProduct() {
        System.out.println("Product Name: " + name + ", Price: $" +
price + ", Quantity: " + quantity);
    }
}

class Inventory {
    private ArrayList<Product> products;

    // Constructor for Inventory
    public Inventory() {
        products = new ArrayList<>();
    }

    // Method to add a product to the inventory
    public void addProduct(Product product) {
        products.add(product);
        System.out.println("Product added to inventory: " +
product.name);
    }

    // Method to display all products in the inventory
    public void displayInventory() {
        System.out.println("\n--- Inventory Details ---");
        for (Product product : products) {
            product.displayProduct();
        }
    }
```

```
}


public class Lab_program1b {
    public static void main(String[] args) {
        // Create an inventory
        Inventory storeInventory = new Inventory();

        // Create products using different constructors
        Product defaultProduct = new Product(); // Using default
constructor
        Product specificProduct = new Product("Laptop", 1200.50, 5);
// Using parameterized constructor
        Product copiedProduct = new Product(specificProduct); // Using
copy constructor

        // Add products to the inventory
        storeInventory.addProduct(defaultProduct);
        storeInventory.addProduct(specificProduct);
        storeInventory.addProduct(copiedProduct);

        // Display all products in the inventory
        storeInventory.displayInventory();
    }
}
```

PROGRAM 2:

Aim: Demonstrating creation of java classes, objects, constructors, declaration and

initialization of variables.

Program: Create a Java class called Student with the following details as variables

within it.

- **USN**
- **Name**
- **Branch**
- **Phone**

Write a Java program to create n Student objects and print the USN, Name, Branch,

and Phone of these objects with suitable headings.

```
public class Student1
{
String usn, name, branch;
long ph;
Student1()
{
usn = name = branch = "no value";
ph = 0;
}
void read_data(String u, String n, String b, long p)
{
usn = u;
name = n;
branch = b;
ph =p;
}
void display()
{
System.out.println(usn+"\t\t"+name+"\t\t"+branch+"\t\t"+ ph);
}
```

```
public static void main(String args[]) throws Exception
{
String u, n, b;
long p;
int no;
BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
System.out.println("Enter number of records");
no = Integer.parseInt(br.readLine());
Student1[] s = new Student1[no];
for(int i=0; i<s.length;i++)
{
System.out.println("Enter " + (i + 1) + " Student record");
s[i] = new Student1();
System.out.println("Enter student USN");
u = br.readLine();
System.out.println("Enter student Name");
n = br.readLine();
System.out.println("Enter student Branch");
b = br.readLine();
System.out.println("Enter student Phone number");
p = Long.parseLong(br.readLine());
s[i].read_data(u, n, b, p);
}
System.out.println("USN \t NAME \t BRANCH \t PHONE NO");
 for (int i=0;i<s.length;i++)
 s[i].display();
 }
}
```

```
OUTPUT:
Enter number of records
3
Enter 1 Student record
```

```
Enter student USN
CG001
Enter student Name
ABHINAV
Enter student Branch
CSD
Enter student Phone number
8976543423
Enter 2 Student record
Enter student USN
CG002
Enter student Name
BHAVANA
Enter student Branch
CSD
Enter student Phone number
9876564534
Enter 3 Student record
Enter student USN
CG003
Enter student Name
HARSHITHA
Enter student Branch
CSD
Enter student Phone number
9876231254
USN NAME BRANCH PHONE NO
CG001 ABHINAV CSD 8976543423
CG002 BHAVANA CSD 9876564534
CG003 HARSHITHA CSD 9876231254
```

PROGRAM- 03

Aim: Introduce concepts of method overloading, constructor overloading, overriding.

Program:

(A). Write a java program demonstrating Method overloading.

(B). Write a java program demonstrating Constructor overloading.

Method overloading

```
public class Method_overloading {
 //adding two integer numbers
 int add(int a, int b)
 {
 int sum = a+b;
 return sum;
 }
 //adding three integer numbers
 int add(int a, int b, int c)
 {
 int sum = a+b+c;
 return sum;
 }
 double add(double a, double b)
 {
 double sum = a+b;
 return sum;
 }
public static void main(String args[])
 {
 Method_overloading obj = new Method_overloading();
 System.out.println(obj.add(10, 20));
 System.out.println(obj.add(10, 20, 30));
 System.out.println(obj.add(10.4,20.3));
 }
}
```

OUTPUT:

```
30
60
30.700000000000003
```

3(B). Write a java program demonstrating Constructor overloading.

Constructor overloading

```
public class Constructor_overloading
{
//instance variables of the class
int id;
String name;

Constructor_overloading ()
{
System.out.println("This a default constructor");
}

Constructor_overloading (int i, String n)
{
id = i;
name = n;
}

public static void main(String[] args)
{
//object creation
Constructor_overloading s1 = new Constructor_overloading ();
System.out.println("\nDefault Constructor values: \n");
System.out.println("Student Id : "+s1.id + "\nStudent Name :
"+s1.name);

System.out.println("\nParameterized Constructor values: \n");
```

```
Constructor_overloading s2 = new Constructor_overloading (10,
"David");

System.out.println("Student Id : "+s2.id + "\nStudent Name :
"+s2.name);

}

}
```

OUTPUT:

```
This a default constructor

Default Constructor values:

Student Id : 0

Student Name : null

Parameterized Constructor values:

Student Id : 10

Student Name : David
```

PROGRAM- 04

Aim: Demonstrate the core object-oriented concept of Inheritance, polymorphism

Program: Design a super class called Staff with details as StaffId, Name, Phone, and Salary. Extend this class by writing three subclasses namely Teaching (domain, publications), Technical (skills), and Contract (period). Write a Java program to read and display at least 3 staff objects of all three

categories.

PROGRAM: WITH INPUT GIVEN

```
class Staff {

int staffid,phone,salary;

String name;

public Staff(int id , int no, int sal, String na){

staffid=id;

phone=no;

salary=sal;

name=na;

}
```

```
void display(){
System.out.println("-------------------------------------");
System.out.println("Staff ID:"+ " "+ staffid);
System.out.println("Staff Phone number:" + " "+ phone);
System.out.println("Staff Salary:" +" "+ salary);
System.out.println("Staff Name:" +" "+ name);
}
}
class Teaching extends Staff {
String domain;
int no_of_publications;
public Teaching(int id, int no, int sal, String na,String d,int nop){
super(id,no,sal,na);
domain=d;
no_of_publications=nop;
}
void Tdisplay(){
System.out.println("-------------------------------------");
System.out.println("\nTeaching Staff Details\n");
super.display();
System.out.println("Domain :" +" "+domain);
System.out.println("No_of_publications:"+" "+no_of_publications);
}
}
class Technical extends Staff{
String skills;
public Technical(int id , int no, int sal, String na,String sk){
super(id,no,sal,na);
skills=sk;
}
void Tedisplay(){
System.out.println("-------------------------------------");
```

```
System.out.println("Technical Staff Details");
super.display();
System.out.println("Skills :" + " "+skills);
}
}
class Contract extends Staff{
int period;
public Contract(int id , int no, int sal, String na,int pd){
super(id,no,sal,na);
period=pd;
}
void Cdisplay(){
System.out.println("-------------------------------------");
System.out.println("Contract Staff Details");
super.display();
System.out.println("ContractPeriod:" + " "+period + "years");
}
}
class Multilevel
{
public static void main(String args[]){
Teaching t1=new Teaching(11,998765434,31000,"Anil","CSE",10);
Teaching t2=new Teaching(12,996655546,30000,"Anu","ISE",9);
Teaching t3=new Teaching(13,999933442,32000,"Anusha","EEE",8);
t1.Tdisplay();
t2.Tdisplay();
t3.Tdisplay();
Technical te1=new Technical(21,994433221,22000,"Kumar","C");
Technical te2=new Technical(22,998877665,28000,"Krisna","Java");
Technical te3=new Technical(23,991654321,33000,"Kiran","Java");
te1.Tedisplay();
te2.Tedisplay();
```

```
te3.Tedisplay();
Contract ct1=new Contract(31,998765434,35000,"Anil",3);
Contract ct2=new Contract(32,912345678,39000,"Meghana",2);
Contract ct3=new Contract(33,992233445,30000,"Uma",4);
ct1.Cdisplay();
ct2.Cdisplay();
ct3.Cdisplay();
}
}
```

OUTPUT:

```
-------------------------------------
Teaching Staff Details
-------------------------------------
Staff ID: 11
Staff Phone number: 998765434
Staff Salary: 31000
Staff Name: Anil
Domain : CSE
No_of_publications: 10
-------------------------------------
Teaching Staff Details
-------------------------------------
Staff ID: 12
Staff Phone number: 996655546
Staff Salary: 30000
Staff Name: Anu
Domain : ISE
No_of_publications: 9
-------------------------------------
Teaching Staff Details
-------------------------------------
Staff ID: 13
```

Staff Phone number: 999933442

Staff Salary: 32000

Staff Name: Anusha

Domain : EEE

No_of_publications: 8

Technical Staff Details

Staff ID: 21

Staff Phone number: 994433221

Staff Salary: 22000

Staff Name: Kumar

Skills : C

Technical Staff Details

Staff ID: 22

Staff Phone number: 998877665

Staff Salary: 28000

Staff Name: Krisna

Skills : Java

Technical Staff Details

Staff ID: 23

Staff Phone number: 991654321

Staff Salary: 33000

Staff Name: Kiran

Skills : Java

Contract Staff Details

```
Staff ID: 31
Staff Phone number: 998765434
Staff Salary: 35000
Staff Name: Anil
ContractPeriod: 3years
-------------------------------------
Contract Staff Details
-------------------------------------
Staff ID: 32
Staff Phone number: 912345678
Staff Salary: 39000
Staff Name: Meghana
ContractPeriod: 2years
-------------------------------------
Contract Staff Details
-------------------------------------
Staff ID: 33
Staff Phone number: 992233445
Staff Salary: 30000
Staff Name: Uma
ContractPeriod: 4years
```

PROGRAM: TAKING THE INPUT FROM USER

```
import java.util.*;
class Staff {
 Scanner in = new Scanner(System.in);
 String StaffId, Name, Phone_no;
 double Salary;
 public void get() {
 System.out.println("Enter the Staff Details:");
 System.out.println("Enter the StaffId: ");
 StaffId = in.nextLine();
```

```
 System.out.println("Enter the Name: ");
 Name = in.nextLine();
 System.out.println("Enter the Phone number: ");
 Phone_no = in.nextLine();
 System.out.println("Enter the Satff Salary: ");
 Salary = in.nextDouble();
 }
 public void display() {
 System.out.println("Staff Name " + Name);
 System.out.println("Staff ID " + StaffId);
 System.out.println("Staff Phone number " + Phone_no);
 System.out.println("Staff Salary " + Salary);
 }
}
class Teaching extends Staff {
 String Domain, Publication;
 public void read() {
 System.out.println("Enter Teaching Details: ");
 super.get();
 System.out.println("Enter the Domain and Publication: ");
 Domain = in.nextLine();
 Publication = in.nextLine();
 }
 public void show() {
 System.out.println("Displaying Teaching Details: ");
 super.display();
 System.out.println("Domain: " + Domain + "\nPublication: " +
Publication);
 }
}
class Technical extends Staff {
 String skills;
 public void read() {
```

```
 System.out.println("Enter the details of Technical");
 super.get();
 System.out.println("Enter the skills : ");
 skills = in.nextLine();
 }
 public void show() {
 System.out.println("Displaying Technical Details: ");
 super.display();
 System.out.println("Skills: " + skills);
 }
}
class contract extends Staff {
 int Period;
 public void read() {
 System.out.println("Enter the Contract Details: ");
 super.get();
 System.out.println("Enter the Period: ");
 Period = in.nextInt();
 }
 public void show() {
 System.out.println("Displaying the contract details: ");
 super.display();
 System.out.println("Periods: " + Period);
 }
}
public class StaffDetails {
 public static void main(String[] args) {
 int n;
 Scanner input = new Scanner(System.in);
 System.out.println("Enter the number of Staff: ");
 n = input.nextInt();
 Teaching[] teach = new Teaching[n];
```

```
 Technical[] tech = new Technical[n];
 contract[] cont = new contract[n];
 for (int i = 0; i < n; i++) {
 teach[i] = new Teaching();
 tech[i] = new Technical();
 cont[i] = new contract();
 teach[i].read();
 tech[i].read();
 cont[i].read();
 }
 for (int i = 0; i < n; i++) {
 teach[i].display();
 tech[i].display();
 cont[i].display();
 }
 input.close();
 }
}
```

OUTPUT:

```
Enter the number of Staff:
2
Enter Teaching Details:
Enter the Staff Details:
Enter the StaffId:
12
Enter the Name:
Anil
Enter the Phone number:
6789653423
Enter the Satff Salary:
45000
Enter the Domain and Publication:
```

IoT

Enter the details of Technical

Enter the Staff Details:

Enter the StaffId:

13

Enter the Name:

Bindu

Enter the Phone number:

7890765435

Enter the Satff Salary:

27000

Enter the skills :

Enter the Contract Details:

Enter the Staff Details:

Enter the StaffId:

14

Enter the Name:

Satya

Enter the Phone number:

6789542312

Enter the Satff Salary:

20000

Enter the Period:

12 months

Enter Teaching Details:

Enter the Staff Details:

Enter the StaffId:

11

Enter the Name:

Harshitha

Enter the Phone number:

9786543241

```
Enter the Satff Salary:
38000
Enter the Domain and Publication:
Networks
Enter the details of Technical
Enter the Staff Details:
Enter the StaffId:
10
Enter the Name:
Preethi
Enter the Phone number:
7896543212
Enter the Satff Salary:
28000
Enter the skills :
Enter the Contract Details:
Enter the Staff Details:
Enter the StaffId:
16
Enter the Name:
Vani
Enter the Phone number:
8976567690
Enter the Satff Salary:
20000
Enter the Period:
15 months
Staff Name Anil
Staff ID 12
Staff Phone number 6789653423
Staff Salary 45000.0
Staff Name Bindu
```

```
Staff ID 13
Staff Phone number 7890765435
Staff Salary 27000.0
Staff Name Satya
Staff ID 14
Staff Phone number 6789542312
Staff Salary 20000.0
Staff Name Harshitha
Staff ID 11
Staff Phone number 9786543241
Staff Salary 38000.0
Staff Name Preethi
Staff ID 10
Staff Phone number 7896543212
Staff Salary 28000.0
Staff Name Vani
Staff ID 16
Staff Phone number 8976567690
Staff Salary 20000.0
```

PROGRAM- 05

Aim: Exception handling in java, introduction to throwable class, throw, throws, finally.

Program: Write a Java program to read two integers a and b. Compute a/b and print,
when b is not zero. Raise an exception when b is equal to zero.

Exception_handle,java

```
import java.util.Scanner;
class Exception_handle
{
 public static void main(String[] args)
 {
  int a,b,result;
  Scanner input =new Scanner(System.in);
```

```
  System.out.println("Input two integers");
  a=input.nextInt();
  b=input.nextInt();
  try
  {
  result=a/b;
  System.out.println("Result="+result);
  }
  catch(ArithmeticException e)
  {
  System.out.println("exception caught: Divide by zero error"+e);
  }
 }
}
```

OUTPUT:

```
Input two integers
4
2
Result=2

Input two integers
4
0
exception caught: Divide by zero errorjava.lang.ArithmeticException: /
by zero
```

PROGRAM- 06

Aim: Introduce the concept of Abstraction, packages.

Program: Develop a java application to implement currency converter (Dollar to INR, EURO to INR, Yen to INR and vice versa), distance converter (meter to KM, miles to KM and vice versa), time converter (hours to minutes, seconds and vice versa) using packages.

Converter.java

```
import java.util.*;
import com.mycompany.converter.Currency;
import com.mycompany.converter.Distance;
import com.mycompany.converter.time;

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

```
{
 Scanner s=new Scanner(System.in);
 int choice,ch;
 Currency c=new Currency();
 Distance d=new Distance();
 time t=new time();
 do
 {
  System.out.println("1.dollar to rupee ");
  System.out.println("2.rupee to dollar ");
  System.out.println("3.Euro to rupee ");
  System.out.println("4.rupee to Euro ");
  System.out.println("5.Yen to rupee ");
  System.out.println("6.Rupee to Yen ");
  System.out.println("7.Meter to kilometer ");
  System.out.println("8.kilometer to meter ");
  System.out.println("9.Miles  to kilometer ");
  System.out.println("10.kilometer to miles");
  System.out.println("11.Hours to Minutes");
  System.out.println("12.Hours to Seconds");
  System.out.println("13.Seconds to Hours");
  System.out.println("14.Minutes to Hours");
  System.out.println("Enter ur choice");
  choice=s.nextInt();

  switch(choice)
  {
   case 1: c.dollartorupee();
     break;

   case 2: c.rupeetodollar();
     break;

   case 3: c.eurotorupee();
     break;

   case 4: c.rupeetoeuro();
     break;

   case 5: c.yentorupee();
     break;

   case 6: c.rupeetoyen();
     break;

   case 7: d.mtokm();
     break;

   case 8: d.kmtom();
     break;
```

```
    case 9: d.milestokm();
      break;

    case 10:d.kmtomiles();
      break;

    case 11:t.hourstominutes();
      break;

    case 12:t.hourstoseconds();
      break;

    case 13:t.secondstohours();
      break;

    case 14:t.minutestohours();
      break;

   }
   System.out.println("Enter 0 to  quit and 1 to continue ");
   ch=s.nextInt();
  }while(ch==1);
 }

}
```

Currency.java

```
import java.util.Scanner;
public class Currency {
double inr,usd;
double euro,yen;
Scanner in=new Scanner(System.in);
  public void dollartorupee()
{
 System.out.println("Enter dollars to convert into Rupees:");
 usd=in.nextInt();
 inr=usd*67;
 System.out.println("Dollar ="+usd+"equal to INR="+inr);
}
public void rupeetodollar()
{
 System.out.println("Enter Rupee to convert into Dollars:");
 inr=in.nextInt();
 usd=inr/67;
 System.out.println("Rupee ="+inr+"equal to Dollars="+usd);
}
public void eurotorupee()
```

```
 {
  System.out.println("Enter euro to convert into Rupees:");
  euro=in.nextInt();
  inr=euro*79.50;
  System.out.println("Euro ="+euro +"equal to INR="+inr);
 }
 public void rupeetoeuro()
 {
  System.out.println("Enter Rupees to convert into Euro:");
  inr=in.nextInt();
  euro=(inr/79.50);
  System.out.println("Rupee ="+inr +"equal to Euro="+euro);
 }
 public void yentorupee()
 {
  System.out.println("Enter yen to convert into Rupees:");
  yen=in.nextInt();
  inr=yen*0.61;
  System.out.println("YEN="+yen +"equal to INR="+inr);
 }
 public void rupeetoyen()
 {
  System.out.println("Enter Rupees to convert into Yen:");
  inr=in.nextInt();
  yen=(inr/0.61);
  System.out.println("INR="+inr +"equal to YEN"+yen);
 }
}
```

Distance.java

```
import java.util.Scanner;
class Distance {
double km,m,miles;
Scanner sc = new Scanner(System.in);
  public void kmtom()
{
 System.out.print("Enter in km ");
 km=sc.nextDouble();
 m=(km*1000);
 System.out.println(km+"km" +"equal to"+m+"metres");
}
public void mtokm()
{
 System.out.print("Enter in meter ");
 m=sc.nextDouble();
 km=(m/1000);
 System.out.println(m+"m" +"equal to"+km+"kilometres");
}
```

```
 public void milestokm()
 {
  System.out.print("Enter in miles");
  miles=sc.nextDouble();
  km=(miles*1.60934);
  System.out.println(miles+"miles" +"equal to"+km+"kilometres");
 }
 public void kmtomiles()
 {
  System.out.print("Enter in km");
  km=sc.nextDouble();
  miles=(km*0.621371);
  System.out.println(km+"km" +"equal to"+miles+"miles");
 }
}
```

Time.java

```
import java.util.Scanner;
public class time {
int hours,seconds,minutes;
 int input;
 Scanner sc = new Scanner(System.in);
 public void secondstohours()
 {
  System.out.print("Enter the number of seconds: ");
  input = sc.nextInt();
  hours = input / 3600;
  minutes = (input % 3600) / 60;
  seconds = (input % 3600) % 60;
  System.out.println("Hours: " + hours);
  System.out.println("Minutes: " + minutes);
  System.out.println("Seconds: " + seconds);
 }
 public void minutestohours()
 {
  System.out.print("Enter the number of minutes: ");
  minutes=sc.nextInt();
  hours=minutes/60;
  minutes=minutes%60;
  System.out.println("Hours: " + hours);
  System.out.println("Minutes: " + minutes);
 }
 public void hourstominutes()
 {
  System.out.println("enter the no of hours");
  hours=sc.nextInt();
  minutes=(hours*60);
  System.out.println("Minutes: " + minutes);
 }
 public void hourstoseconds()
```

```
 {
  System.out.println("enter the no of hours");
  hours=sc.nextInt();
  seconds=(hours*3600);
  System.out.println("Minutes: " + seconds);
 }
}
```

OUTPUT:

```
1.dollar to rupee
2.rupee to dollar
3.Euro to rupee
4.rupee to Euro
5.Yen to rupee
6.Rupee to Yen
7.Meter to kilometer
8.kilometer to meter
9.Miles  to kilometer
10.kilometer to miles
11.Hours to Minutes
12.Hours to Seconds
13.Seconds to Hours
14.Minutes to Hours
Enter ur choice
2
Enter Rupee to convert into Dollars:
45
Rupee =45.0equal to
Dollars=0.6716417910447762
Enter 0 to  quit and 1 to continue
1
1.dollar to rupee
2.rupee to dollar
3.Euro to rupee
4.rupee to Euro
5.Yen to rupee
6.Rupee to Yen
7.Meter to kilometer
8.kilometer to meter
9.Miles  to kilometer
10.kilometer to miles
11.Hours to Minutes
12.Hours to Seconds
13.Seconds to Hours
14.Minutes to Hours
Enter ur choice
9
Enter in miles 45
45.0milesequal to72.4203kilometres
```

```
Enter 0 to  quit and 1 to continue
1
1.dollar to rupee
2.rupee to dollar
3.Euro to rupee
4.rupee to Euro
5.Yen to rupee
6.Rupee to Yen
7.Meter to kilometer
8.kilometer to meter
9.Miles  to kilometer
10.kilometer to miles
11.Hours to Minutes
12.Hours to Seconds
13.Seconds to Hours
14.Minutes to Hours
Enter ur choice
13
Enter the number of seconds: 123456
Hours: 34
Minutes: 17
Seconds: 36
Enter 0 to  quit and 1 to continue
0
```

PROGRAM- 07

Aim: Demonstrate creation of threads using Thread class and Runnable interface, multithreaded programming.

Program: Write a Java program that implements a multi-thread application that has three threads. Frist thread generates a random integer for every 1 seconds; second thread computes the square of the number and prints; third thread will print the value of cube of the number.

Thread_program.java

```
import java.util.Random;
```

```
class Square extends Thread
{
  int x;

  Square(int n)
  {
   x = n;
  }

  public void run()
  {
   int sqr = x * x;
   System.out.println("Square of " + x + " = " + sqr );
  }
}

class Cube extends Thread
{
  int x;

  Cube(int n)
  {
   x = n;
  }

  public void run()
  {
   int cub = x * x * x;
   System.out.println("Cube of " + x + " = " + cub );
  }
}

class Number extends Thread
{
 public void run()
 {
  Random random = new Random();
  for(int i =0; i<10; i++)
  {
   int randomInteger = random.nextInt(30);
   System.out.println("Random Integer generated : " + randomInteger);
   Square s = new Square(randomInteger);
   s.start();
   Cube c = new Cube(randomInteger);
   c.start();
   try
   {
    Thread.sleep(1000);
   }
   catch (InterruptedException ex)
```

```
    {
     System.out.println(ex);
    }
   }
   }
}
public class Thread_program
{
  public static void main(String args[])
  {
   Number n = new Number();
   n.start();
  }
}
```

OUTPUT:

```
Random Integer generated : 0
Square of 0 = 0
Cube of 0 = 0
Random Integer generated : 23
Square of 23 = 529
Cube of 23 = 12167
Random Integer generated : 9
Cube of 9 = 729
Square of 9 = 81
Random Integer generated : 22
Square of 22 = 484
Cube of 22 = 10648
Random Integer generated : 10
Square of 10 = 100
Cube of 10 = 1000
Random Integer generated : 27
Cube of 27 = 19683
Square of 27 = 729
Random Integer generated : 23
Square of 23 = 529
Cube of 23 = 12167
Random Integer generated : 15
Square of 15 = 225
Cube of 15 = 3375
Random Integer generated : 17
Square of 17 = 289
Cube of 17 = 4913
Random Integer generated : 15
Square of 15 = 225
Cube of 15 = 3375
```

PROGRAM- 08

Aim: Introduce java Collections.

Program: Write a program to perform string operations using Array List. Write functions for the following:

a. Append- add at end

b. Insert- add at particular index

c. Search

d. List all string starts with given letter.

Collection_program.java

```
import java.util.*;
import java.io.*;
import java.lang.*;

public class Collection_program
{
 public static void main(String args[])throws IOException
 {
 ArrayList<String> obj = new ArrayList<String>();
 DataInputStream in=new DataInputStream(System.in);
 int c,ch;
 int i,j;
 String str,str1;
 do
 {
  System.out.println("STRING MANIPULATION");
  System.out.println("******************************");
  System.out.println(" 1. Append at end \n 2.Insert at particular
index \n 3.Search \n 4.List
string that starting with letter\n 5.display\n");
  System.out.println("Enter the choice ");
  c=Integer.parseInt(in.readLine());
  switch(c)
  {
  case 1:
  {
   System.out.println("Enter the string ");
   str=in.readLine();
   obj.add(str);
   break;
  }
  case 2:
  {
   System.out.println("Enter the string ");
   str=in.readLine();
   System.out.println("Specify the index/position to insert");
   i=Integer.parseInt(in.readLine());
   obj.add(i-1,str);
   System.out.println("The array list has following elements:"+obj);
   break;
  }
  case 3:
  {
```

```
   System.out.println("Enter the string to search ");
   str=in.readLine();
   j=obj.indexOf(str);
   if(j==-1)
    System.out.println("Element not found");
   else
    System.out.println("Index of:"+str+"is"+j);

   break;
  }
  case 4:
  {
  System.out.println("Enter the character to List string that starts
with specified character");
   str=in.readLine();
   for(i=0;i<(obj.size()-1);i++)
   {
    str1=obj.get(i);
    if(str1.startsWith(str))
    {
     System.out.println(str1);
    }
   }
   break;
  }
  case 5:
  {
   System.out.println("The array list has following elements:"+obj);
   break;
  }
 }
  System.out.println("enter 0 to break and 1 to continue");
  ch=Integer.parseInt(in.readLine());
 }while(ch==1);
 }
}
```

OUTPUT:

```
1. STRING MANIPULATION
*****************************
 1. Append at end
 2.Insert at particular index
 3.Search
 4.List string that starting with letter
 5.display

Enter the choice
1
Enter the string
hi
```

```
enter 0 to break and 1 to continue
1
2. STRING MANIPULATION
*****************************
 1. Append at end
 2.Insert at particular index
 3.Search
 4.List string that starting with letter
 5.display

Enter the choice
1
Enter the string
hello
enter 0 to break and 1 to continue
1
3. STRING MANIPULATION
*****************************
 1. Append at end
 2.Insert at particular index
 3.Search
 4.List string that starting with letter
 5.display

Enter the choice
5
The array list has following elements:[hi, hello]
enter 0 to break and 1 to continue
1
4. STRING MANIPULATION
*****************************
 1. Append at end
 2.Insert at particular index
 3.Search
 4.List string that starting with letter
 5.display

Enter the choice
2
Enter the string
good
Specify the index/position to insert
1
The array list has following elements:[good, hi,
hello]
enter 0 to break and 1 to continue
1
5. STRING MANIPULATION
*****************************
 1. Append at end
 2.Insert at particular index
```

```
 3.Search
 4.List string that starting with letter
 5.display

Enter the choice
5
The array list has following elements:[good, hi,
hello]
enter 0 to break and 1 to continue
1
6. STRING MANIPULATION
*****************************
 1. Append at end
 2.Insert at particular index
 3.Search
 4.List string that starting with letter
 5.display

Enter the choice
3
Enter the string to search
hi
Index of:hiis1
enter 0 to break and 1 to continue
1
7. STRING MANIPULATION
*****************************
 1. Append at end
 2.Insert at particular index
 3.Search
 4.List string that starting with letter
 5.display

Enter the choice
4
Enter the character to List string that starts with
specified character
h
hi
enter 0 to break and 1 to continue
```

PROGRAM-09

Aim: Introduce File operations in java.

Program: Write a java program that reads a file name from the user, displays information about whether the file exists, whether the file is readable, or writable, the type of file and the length of the file in bytes

FileDetails.java

```
import java.io.*;
import java.util.*;

class FileDetails
{
 public static void main(String[] args)
 {
  Scanner input = new Scanner(System.in);
  System.out.print("Enter the name of the file:");
  String file_name = input.nextLine();
  File f = new File(file_name);

        if(f.exists())
         System.out.println("The file " +file_name+ " exists");
         else
         System.out.println("The file " +file_name+ " does not
exist");

        if(f.exists())
        {
         if(f.canRead())
          System.out.println("The file " +file_name+ " is readable");
          else
          System.out.println("The file " +file_name+ " is not
readable");

            if(f.canWrite())
             System.out.println("The file " +file_name+ " is
writeable");
             else
             System.out.println("The file " +file_name+ " is not
writeable");
                System.out.println("The file type is: "
+file_name.substring(file_name.indexOf('.')+1));
                System.out.println("The Length of the file:"
+f.length());
          }
     }
}
```

OUTPUT:

```
Enter the name of the file:C:\Users\CSD-LAB3\Desktop\VTU Registration
```

```
The file C:\Users\CSD-LAB3\Desktop\VTU Registration exists
The file C:\Users\CSD-LAB3\Desktop\VTU Registration is readable
The file C:\Users\CSD-LAB3\Desktop\VTU Registration is writeable
The file type is: C:\Users\CSD-LAB3\Desktop\VTU Registration
The Length of the file:24576

Enter the name of the
file:C:\Users\CSD-LAB3\Documents\NetBeansProjects
The file C:\Users\CSD-LAB3\Documents\NetBeansProjects exists
The file C:\Users\CSD-LAB3\Documents\NetBeansProjects is readable
The file C:\Users\CSD-LAB3\Documents\NetBeansProjects is writeable
The file type is: C:\Users\CSD-LAB3\Documents\NetBeansProjects
The Length of the file:4096
```

PROGRAM- 10

Program: Write a program to generate the resume. Create 2 Java classes Teacher (data: personal information, qualification, experience, achievements) and Student (data: personal information, result, discipline) which implements the java interface Resume with the method biodata().

```
interface Resume
{
 public void bio_data();
}

class Teacher implements Resume
{
 String personal_information="suma";
 String qualification="BE";
 String experience="10";
 String achivments="Gold medal";

 public void bio_data()
 {
  System.out.println("personal_information:\t"+personal_information);
  System.out.println("qualification:\t"+qualification);
  System.out.println("experience:\t"+experience);
  System.out.println("achivments:\t"+achivments);
 }
}

class Student
{
```

```
 String personal_information="uma";
 String result="80";
 String discipline="ISE";

 public void bio_data()
 {
  System.out.println("personal_information:\t"+personal_information);
  System.out.println("result:\t"+result);
  System.out.println("discipline:\t"+discipline);
 }
}

class Interface_main
{
 public static void main(String[] args)
 {
  Teacher teach=new Teacher();
  teach.bio_data();

  Student stud=new Student();
  stud.bio_data();
 }
}
```

OUTPUT:

```
personal_information: suma
qualification: BE
experience: 10
achivments: Gold medal
personal_information: uma
result: 80
discipline: ISE
```