

Program 1

Java program to print an Integer (Entered by the user)

Algorithm

Step 1 : Start

Step 2 : Create a reader instance to take input

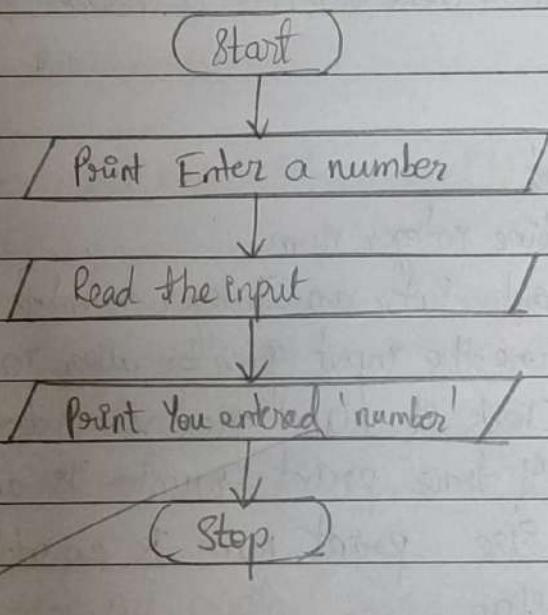
Step 3 : Print Enter a number

Step 4 : Read the number entered by user

Step 5 : Print You entered (the number entered by user)

Step 6 : Stop

Flowchart



```
import java.util.Scanner;
public class HelloWorld {
    public static void main (String[] args) {
        Scanner reader = new Scanner(System.in);
        System.out.print ("Enter a number: ");
        int number = reader.nextInt();
        System.out.println ("You entered: " + number);
    }
}
```

O/P: Enter a number: 4
You entered: 4

2) Write a program to check whether a number is Even or Odd

Algorithm

Step 1: Start

Step 2: Define integer num

Step 3: Print Enter an Integer number

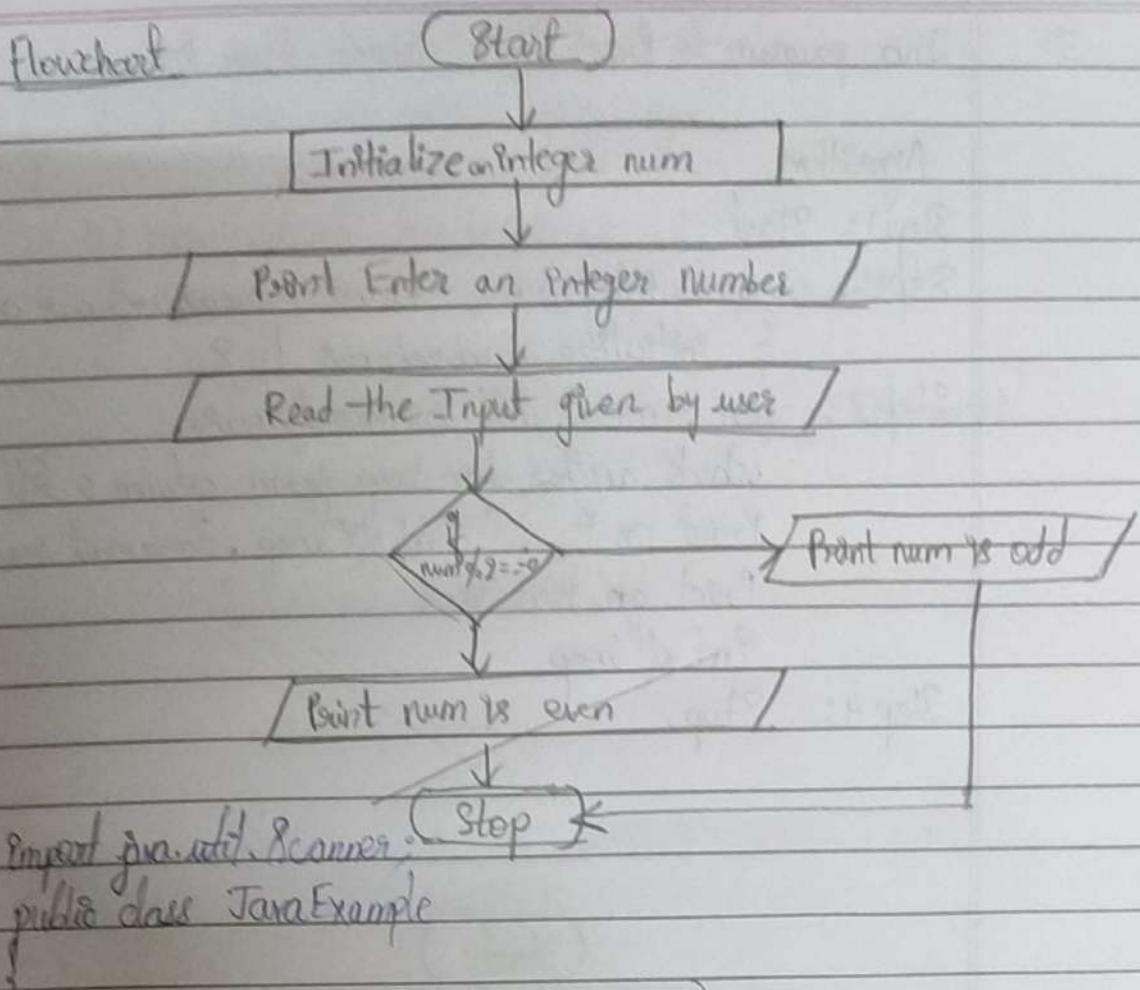
Step 4: Store the input given by user in num

Step 5: Check if num gives remainder 0 when divided by 2

Step 6: If true print number is an even number

Step 7: Else print num is an odd number

Step 8: Stop



```

public static void main (String args[])
{
    int num;
    Scanner input = new Scanner (System.in);
    num = input.nextInt();
    if (num % 2 == 0)
        System.out.println (num + " is an even number.");
    else
        System.out.println (num + " is an odd number.");
}
    
```

dp: Enter an Integer : 8
8 is an even no

3> Java program to Print Right Triangle Star Pattern

Algorithm

Step1: Start

Step2: Define int row, column , numberofrows as integer
& initialize numberofrows to 8.

Step3: for row = 0 till numberofrows

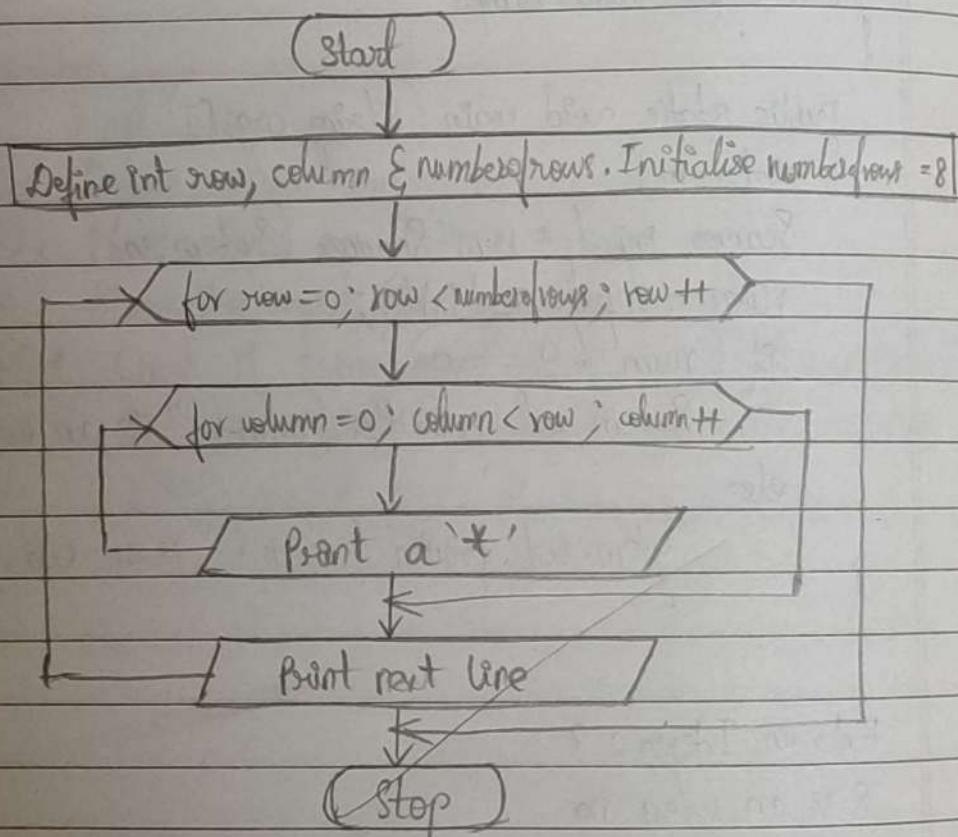
 check another for loop from column = 0 till row

 Point an * . End of loop . Increment row by 1

 Point an next line

 End of loop

Step4: Stop.



```

public class JavaExample
{
    public static void main [String args[]]
    {
        int row, column, numberofrows = 8;
        for (row = 0; row < numberofrows; row++)
        {
            for (column = 0; column <= row; column++)
            {
                System.out.print ("* ");
            }
            System.out.println ();
        }
    }
}

```

D/P:

```

*
**
***
*****
*****
*****
*****
***** *

```

4) Java program to Find Quotient & Remainder

Algorithm

Step1: Start

Step2: Initialize num1 to 15, num2 to 2. Define them as integers

Step3: Operate quotient = num1 / num2;

Step4: Operate Remainder = num1 % num2

Step5: Print Quotient value

Step6: Print Remainder value

Step7: Stop

Start

Define & Initialize num1 = 15 & num2 = 2

quotient = num1 / num2

remainder = num1 % num2

Print the quotient value

Print the remainder value

Stop

```
public class QANDR {  
    public static void main (String[] args) {  
        int num1 = 15, num2 = 2;  
        int quotient = num1 / num2;  
        int remainder = num1 % num2;  
        System.out.println ("Quotient is: " + quotient);  
        System.out.println ("Remainder is: " + remainder);  
    }  
}
```

O/P: Quotient is: 7

Remainder is: 1

57

5) Java Program to Multiply Two numbers

```

public class Demo {
    public static void main (String[] args) {
        Scanner scan = new Scanner (System.in);
        System.out.println ("Enter first number : ");
        int num1 = scan.nextInt();
        System.out.println ("Enter second number : ");
        int num2 = scan.nextInt();
        int num2 = scan.nextInt();
        scan.close();
        int product = num1 * num2;
        System.out.println ("Output : " + product);
    }
}

```

O/P:

```

Enter first no: 6
Enter second no: 9
Output: 54

```

Algorithm

Step 1: Start

Step 2: Print Enter first number

Step 3: Read input for int num1 from user

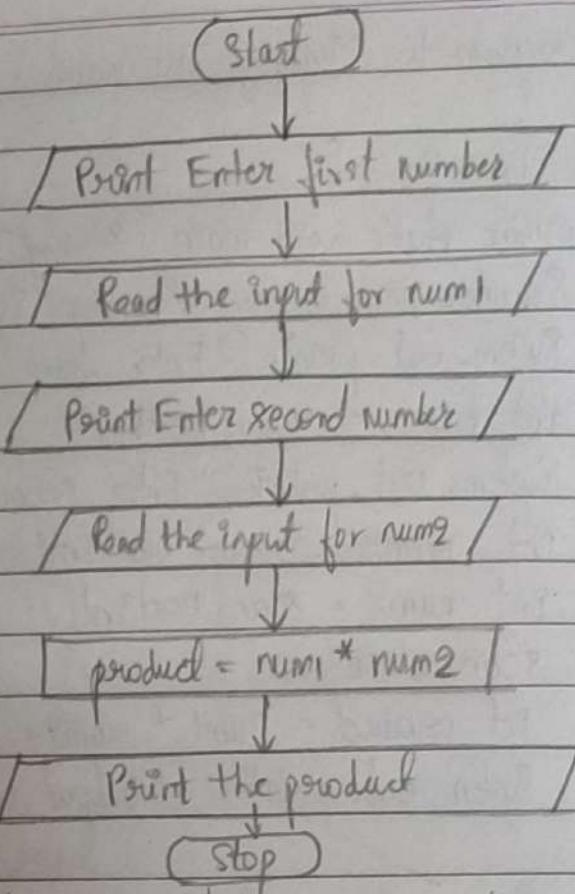
Step 4: Print Enter second number

Step 5: Read input for int num2

Step 6: Calculate product = num1 * num2

Step 7: Print the product value

Step 8: Stop



6) Swap 2 numbers using temporary variable

```
public class Swapnumbers {
```

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

```
        float first = 1.2f, second = 2.45f;
```

```
        System.out.println ("-- Before swap --");
```

```
        System.out.println ("First number = " + first);
```

```
        System.out.println ("Second number = " + second);
```

```
        float temporary = first;
```

```
        first = second;
```

```
        second = temporary;
```

```
        System.out.println ("-- After swap --");
```

```
        System.out.println ("First number = " + first);
```

```
        System.out.println ("Second number = " + second);
```

Algorithm

Step1: Start

Step2: Define first & second as float datatype & initialize to 1.2af and 2.45f respectively.

Step3: Print statement Before Swap

Step4: Print first number

Step5: Print second number

Step6: Create a variable temporary of float datatype

Step7: Store variable first in temp

Step8: Store second in first

Step9: Store temp in second

Step10: Print the statement After Swap

Step11: Print first number : first

Step12: Print second number

Step13: Stop.

(Start)

Define & initialise first = 1.2af & second = 2.45f.

Point 'Before Swap'

Point 'first number'

Point 'second number'

temp = first

first = second

second = temp

```
C:\Users\STUDENT\Desktop\1bm22cs001>java JavaExample2
Enter an integer : 8
8 is an even no
Aakanksha V R 1BM22CS001
```

```
C:\Users\STUDENT\Desktop\1bm22cs001>java JavaEx1
```

```
*  
* *  
* * *  
* * * *  
* * * * *  
* * * * * *  
* * * * * * *
```

```
Aakanksha V R 1BM22CS001
```

```
C:\Users\STUDENT\Desktop\lbn22cs001>javac HelloWorld1.java
```

```
C:\Users\STUDENT\Desktop\lbn22cs001>java HelloWorld1
```

```
Enter a number: 4
```

```
You entered:4
```

```
C:\Users\STUDENT\Desktop\1bm22cs001>java QANDR
Quotient is: 7
Remainder is: 1
Aakanksha 1BM22CS001
```

```
C:\Users\STUDENT\Desktop\1bm22cs001>java QANDR
Quotient is: 7
Remainder is: 1
Aakanksha 1BM22CS001
```

```
C:\Users\STUDENT\Desktop\IBM22CS001>java Demo  
Enter first no: 6  
Enter second no: 9  
Output: 54  
Aakanksha IBM22CS001
```

```
C:\Users\STUDENT\Desktop\1bm22cs001>java Swapnos
--Before swap--
First number= 1.2
Second number= 2.45
--After swap--
First number= 2.45
Second number= 1.2
Aakanksha 1BM22CS001
```

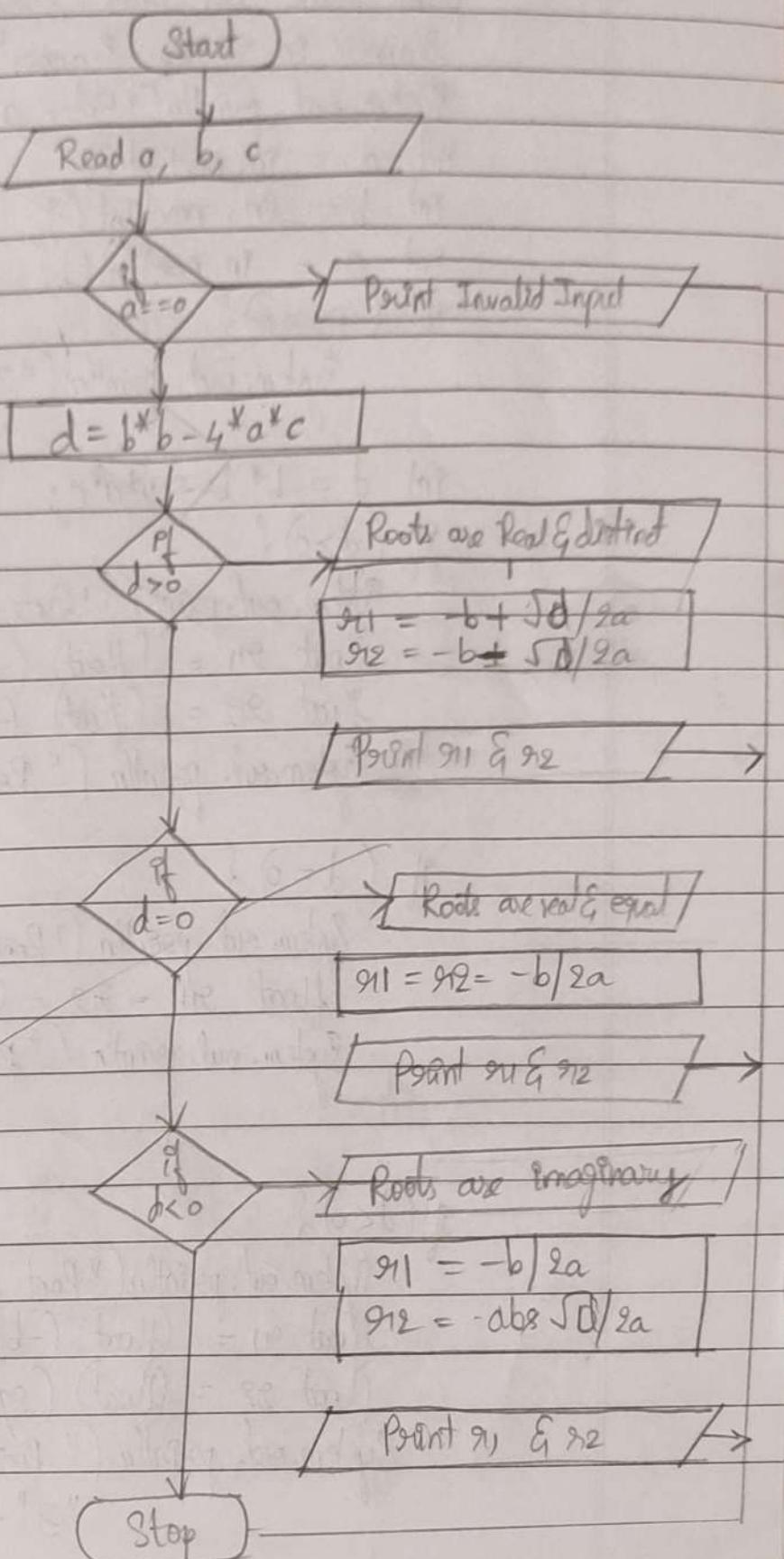
Lab Program 1

Date 22/12
Page

a)

Develop a Java program that prints all real solutions to quadratic equation $ax^2 + bx + c = 0$

Read in a, b, c and use the quadratic formula. If the discriminant $b^2 - 4ac > 0$ is negative, display a message stating that there are no real solutions.



```

import java.util.Scanner;
import static java.lang.Math.sqrt;
import static java.lang.Math.abs;

```

```

public class QF {
    public static void main (String [] args) {
        Scanner in = new Scanner (System.in);
        System.out.println ("Enter a, b & c:");
        int a = in.nextInt ();
        int b = in.nextInt ();
        int c = in.nextInt ();
        if (a == 0) {
            System.out.println ("Invalid Input for a");
        }
    }
}

```

$$\text{int } d = b * b - 4 * a * c;$$

```

if (d > 0) {

```

System.out.println ("Roots are real & distinct");

$$\text{float } r_1 = (\text{float}) (-b + \sqrt{d}) / 2 * a;$$

$$\text{float } r_2 = (\text{float}) (-b - \sqrt{d}) / 2 * a;$$

System.out.println ("Roots are " + r1 + " & " + r2);

```

if (d == 0) {

```

System.out.println ("Roots are real & equal");

$$\text{float } r_1 = r_2 = (\text{float}) (-b / 2 * a);$$

System.out.println ("Roots are " + r1 + " & " + r2);

```

if (d < 0) {

```

System.out.println ("Roots are imaginary");

$$\text{float } r_1 = (\text{float}) (-b / 2 * a);$$

$$\text{float } r_2 = (\text{float}) (\sqrt{-d}) / 2 * a;$$

System.out.println ("Roots are " + r1 + " + " + r2 +
 "i" + r1 + " - " + r2);

dp1 Enter a, b, c

0

2

3

Invalid input for a

dp2 Enter a, b, c

2

7

1

Roots are real and distinct

Roots are -0.5968758 ± 13.403124

dp3 Enter a, b, c

2

4

2

Roots are real and equal

Roots are -4.0 ± 4.0

dp4 Enter a, b, c

5

2

3

Roots are P imaginary

Roots are $-1.0 + 21.414i \pm 1.0 - 21.414i$

22/12/23

```
C:\Users\bmsce\Desktop\1BM22CS001>java QE
```

```
Enter a,b,c
```

```
0
```

```
2
```

```
3
```

```
Aakanksha V R 1BM22CS001
```

```
Invalid input for a
```

```
C:\Users\bmsce\Desktop\IBM22CS001>java QE
```

```
Enter a,b,c
```

```
2
```

```
7
```

```
1
```

```
Aakanksha V R IBM22CS001
```

```
Roots are real and distinct
```

```
Roots are -0.5963758 & -13.403124
```

```
C:\Users\bmsce\Desktop\IBM22CS001>java QE
```

```
Enter a,b,c
```

```
1
```

```
2
```

```
3
```

```
Aakanksha V R IBM22CS001
```

```
Roots are imaginary
```

```
Roots are -1.0+i1.4142135 & -1.0-i1.4142135
```

```
C:\Users\bmsce\Desktop\IBM22CS001>java QE
Enter a,b,c
2
4
2
Aakanksha V R IBM22CS001
Roots are real and equal
Roots are -4.0 & -4.0
```

29/12/23

- P2. Develop a Java Program to create a class student with members usn, name, an array credits & an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

Algorithm

Start

- 1) Create a class Student.
- 2) Declare instance variables usn, name of string datatype.
Declare arrays credits & marks of size 8
- 3) Create a method Details.
Read input Name & USN.
Read subject credit and marks scored by student.
- 4) Create a method CalSGPA to calculate SGPA
Initialise totalCredits & sum to 0. Declare variable ans
for p = 0 till p = credits.length
Read credits and marks. Add credits.
Increment i by 1. If i > credits.length exit the loop
if i > Compute gradeValue = (marks[i] / 10) + 1
Create a method
if gradeValue = 11, consider it as 10
else if gradeValue = 4, consider it as 0

$$ans = \frac{\text{totalSum}}{\text{totalCredits}}$$

- 5) Create a method SGPA.
Create an object student of type Student
Call method Details
Print Student USN, Name
Print Call method CalSGPA.
Print SGPA.
- 6) Stop.

Import java.util.Scanner;

```
class Student {
    String rollno;
    String name;
    int[] credits = new int[8];
    int[] marks = new int[8];
```

```
public void details() {
    Scanner input = new Scanner(System.in);
    System.out.println("Enter URN:");
    URN = input.nextLine();
    System.out.println("Enter name:");
    name = input.nextLine();
```

~~System.out.println("Enter details: \n");~~

```
for (int i = 0; i < credits.length; i++) {
    System.out.println("In Enter the credits of subject" +
        (i + 1));
    credits[i] = input.nextInt();
    System.out.println("Enter the marks scored in that
        subject" + (i + 1));
    marks[i] = input.nextInt();}
```

}

public double CalGPA() {

int totalCredits = 0;

int sum = 0;

double ans;

for (int i = 0; i < credits.length; i++) {
 totalCredits += credits[i];
 int gradeValue;
 gradeValue = (marks[i] / 10) + 1;

```
if (gradeValue == 11)  
    {  
        gradeValue = 10;  
    }
```

```
else if (gradeValue <= 4){  
    gradeValue = 0;  
}
```

```
} sum += gradeValue * credits[?];
```

```
ans = (double) sum / (double) totalCredits;
```

```
return ans;
```

```
} } public class SGPA {
```

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

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

```
Student student = new Student ();
```

```
student.Details ();
```

```
System.out.println ("In Student Details.");
```

```
System.out.println ("UIN : " + student.uin);
```

```
System.out.println ("Name : " + student.name);
```

```
double SGPA = student.getSGPA ();
```

```
System.out.println ("SGPA = " + SGPA);
```

2/12/23

Enter USN:

CS001

Enter Name:

Aakanksha.V.R

Enter details:

Enter the credits of subject1:

4

Enter the marks of subject1:

95

Enter the credits of subject2:

4

Enter the marks of subject2:

97

Enter the credits of subject3:

3

Enter the marks of subject3:

90

Enter the credits of subject4:

3

Enter the marks of subject4:

85

Enter the credits of subject5:

3

Enter the marks of subject5:

84

Enter the credits of subject6:

Enter the credits of subject3:

3

Enter the marks of subject3:

90

Enter the credits of subject4:

3

Enter the marks of subject4:

85

Enter the credits of subject5:

3

Enter the marks of subject5:

84

Enter the credits of subject6:

1

Enter the marks of subject6:

99

Enter the credits of subject7:

1

Enter the marks of subject7:

95

Enter the credits of subject8:

1

Enter the marks of subject8:

90

Student details:

USN: CS001

NAME: Aakanksha.V. R

SGPA: 9.7

Program 3

Date 12/1/24
Page

import

Create a class Book which contains 4 members : name, author, price, no. pages.

Include a constructor to set the values for the members.
Include methods to set & get the details of object. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

Algo:

Step1: Start

Step2: Create a class BookDemo.

Step3: Declare instance variables name, author, price, pages.

Step4: Create a no. constructor to instantiate values.

Step5: Create a method to get details for user inputs.

Step6: Create a method to get details to

Step7: Create a method `toString` to display the outputs.

Step8: Class Book is created

Step9: Read New object book is created of ~~book~~ class BookDemo.

Step10: Read inputs for name, author, price, no. pages

Step11: Display the details of all Books.

Step12: Stop.

```
import java.util.Scanner;  
class book  
{  
    String name;  
    String author;  
    double float price;  
    int num pages;
```

```
void setDetails()  
{  
    Scanner in = new Scanner (System.in);  
    System.out.println ("Enter Bookname, author, price,  
    num pages");  
    name = in.next();  
    author = in.next();  
    price = in.next();
```

```
import java.util.Scanner;  
class BookDemo  
{  
    String name;  
    String author;  
    double price;  
    int numpages}
```

```
BookDemo (String name, String author, double price,  
int numpages)  
{  
    this.name = name;  
    this.author = author;  
    this.price = price;  
    this.numpages = numpages } }
```

~~void~~ ~~toString()~~

```

    {
        System.out.println("Name :" + name + " " +
                           "Author :" + author + " " +
                           "Price :" + price +
                           " " + "Noofpages :" + noofpages);
    }

```

~~void~~ ~~setdetails (String name, String author, double price, int noofpages)~~

```

    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.noofpages = noofpages;
    }

```

~~void~~ ~~getdetails (obj b)~~

```

    {
        b.a = b.toString();
    }

```

~~void~~ ~~setdetails ()~~

```

    {
        Scanner in = new Scanner(System.in);
        name = in.nextLine();
        author = in.nextLine();
        price = in.nextDouble();
        noofpages = in.nextInt();
    }

```

~~String~~ ~~getname()~~

```

    {
        return name;
    }

```

~~double~~ ~~getprice()~~

```

    {
        return price;
    }

```

~~void~~ ~~String getauthor()~~

```

    {
        return author;
    }

```

~~int~~ ~~getnoofpages()~~

```

    {
        return noofpages;
    }

```

```

class Book {
    public static void main (String [] args) {
        BookDemo[] book = new BookDemo[n];
        System.out.println ("Enter no of books");
        Scanner in = new Scanner (System.in);
        for (int m = in.nextInt ());
        for (int i = 0; i < n; i++)
            {
                System.out.println ("Enter name, author, price
                    of pages");
                name = in.nextLine();
                author = in.nextLine();
                price = in.nextDouble();
                npages = in.nextInt();
                book[i] = new BookDemo (name, author, price,
                    • adddetails npages)
            }
}

```

```

for (i=0; i < n; i++)
{
    book[i].toString();
    getString();
}

```

```

void toString ()
{
    System.out.println ("Name:");
    getname();
    System.out.println ("Author:" + getauthor());
    System.out.println ("Price:" + getprice());
    System.out.println ("No of pages:" + getnpages());
}

```

```

class Book {
    public static void main (String [] args)
        BookDemo book = new BookDemo ("DEF", "JK");
        BookDemo book2 = new BookDemo ();
        book2.adddetails();
        book2.toString();
}

```

```
C:\Users\skc\Desktop\ooj p>java Book01
Enter number of books
2
Enter bookname, author, price, num_pages
ABC DEF 250 100
Enter bookname, author, price, num_pages
XYZ VWE 355 200
Book details
The book ABC was written by DEF. It consists of 100 pages and costs around $250.0
The book XYZ was written by VWE. It consists of 200 pages and costs around $355.0
Aakanksha V R,1BM22CS001

C:\Users\skc\Desktop\ooj p>
```

Program 4

Date 12/1/24
Page

Develop a Java Program to create an abstract class named Shape that contains two integers & an empty method named printArea(). Provide 3 classes named Rectangle, Triangle and Circle such that each one of class extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of given shape.

Algorithm

- Step 1: Start
- Step 2: Create Abstract class Shape
- Step 3: Create subclass Rectangle which extends to Shape class.
- Step 4: Create set Read inputs for x & y
- Step 5: Create a method Area to display area of rec.
- Step 6: Create subclass Triangle which extends to Shape
- Step 7: Read inputs for height & base
- Step 8: Create a method to find Area.
- Step 9: Create subclass Circle extending to shape
- Step 10: Read input for radius
- Step 11: Create class CalArea
- Step 12: New objects rec, trian & cir are declared.
- Step 13: Call method area for each obj.
- Step 14: Stop.

```
import java.util.Scanner
```

```
abstract class Shape()
{
    abstract void Area(); { int x; int y; }
}
```

```
class Rectangle extends Shape
```

```
{ Scanner in = new Scanner();
System.out.println("Enter length & breadth");
x = in.nextInt();
y = in.nextInt();
```

```
void Area()
```

```
{ System.out.println("Area is "+(x*y)); }
```

```
class Triangle extends Shape
```

```
{ Scanner in = new Scanner();
System.out.println("Enter height & base");
x = in.nextInt();
y = in.nextInt(); }
```

```
void Area()
```

```
{ System.out.println("Area is "+(0.5*x*y)); }
```

```
class Circle extends Shape
```

```
{ Scanner in = new Scanner();
System.out.println("Enter radius");
x = in.nextInt(); }
```

```
void Area()
```

```
{ System.out.println("Area is "+(3.14*x*x)); }
```

class CalArea

{ abstract public static void main () {

 Rectangle rect = new Rectangle (),

 Triangle trian = new Triangle ();

 Circle cir = new Circle ();

 rect.Area ();

 trian.Area ();

 cir.Area (); }

}

0) #P

Enter length, breadth of rectangle

4

5

Area is 28

Enter base and height of triangle

9

6

Area is 27.0

Enter radius of circle

8

Area is 200.96

Aakanksha.V.R 1BN28C8001

C:\Users\skc\Desktop\ooj p>java Shape

Aakanksha V R, 1BM22CS001

enter the radius of the circle

5

area of the circle is78.5

enter the length and breadth of Rectangle

6

7

area of the rectangle is42

enter the base and height of Triangle

9

8

area of the triangle is36.0

Program - 5

19/1

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account & other is current account. The savings account provides compound interest & withdrawal facility but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should maintain a min balance & if balance falls below this level, service charge is imposed.

Create a class Account that stores customer name, account no & type of account. From this derive the classes CurAcct & Savacct to make them more specific to their requirements. Include necessary methods

- a) Accept deposits from customer & update balance.
- b) display balance
- c) Compute deposit interest
- d) Permit withdrawal & update balance.

Check for min balance, impose penalty.

Algorithm

Start

- 1) Create class Account and declare instance variable cname, acno, acctype, balance.
- 2) Create two subclasses of Account Savacct & Curacct.
- 3) Methods for depositing and displaying balances
- * 4) In class Savacct, create a method to calculate compound interest and to withdraw money.
- 5) In class Curacct, create a method to withdraw money if min balance is present and apply a service charge if balance goes below minbalance.
- 6) Create obj of Savacct class and call the methods to get deposit, CalInterest.
- 7) Create obj of Curacct class and call methods withdraw.
- * Compound Interest = amount $(1 + \text{interestRate})^n$
- * B) Stop

```

import java.util.Scanner;
class Account {
    String cname;
    int accno;
    String acctype;
    double balance;

    public Account (String cname, int accno, String acctype)
    {
        this.cname = cname;
        this.accno = accno;
        this.acctype = acctype;
        this.balance = 0.0; }
}

```

```

public void deposit (int amount)
{
    balance += amount;
    System.out.println ("Deposited"); }
}

```

```

public void displayBalance ()
{
    System.out.println ("Balance is: " + balance); }
}

```

```

class Savact extends Account
{
    double IntRate;
}

```

```

public void Savact (String cname, int accno, double IntRate)
{
    super (cname, accno, "Savings");
    this.cname = cname;
    this.accno = accno;
    this.IntRate = IntRate; }
}

```

```

public void CalInterest (int amount, int term)
{
    double Interest = balance * IntRate;
    System.out.println ("The Interest is " + Interest);
}

```

double compInt = ~~balance * Math.pow((1 + interestRate), time)~~
amount
balance += compInt;

deposit (Interest);

System.out.println ("Balance is : " + balance);

{ public void withdraw (int amount) }

{ if (amount < balance)

{ System.out.println ("Amount deducted : " + amount);
balance -= amount;

System.out.println ("Now Balance is : " + balance);

} else

{ System.out.println ("Balance is low"); } }

}

class Current extends Account {

double minBalance;

double serviceCh;

public Current (String cname, int accno, double minBalance, double serviceCh)

{ super (cname, accno); }

this.minBalance = minBalance;

this.serviceCh = serviceCh;

public void withdraw (int amount)

{ if ((balance - amount) > minBalance)

{ balance -= amount; }

System.out.println ("Balance is " + balance);

} else

{ System.out.println ("Money can't be withdrawn
as bal to maintain"); }

} System.out.println ("Pay nonbalance " + nonbalance); }

Q 100
100

class Bank {

 public static void main (String [] args)

}

 Bank ob1 = new Bank ("Neha", 123456, 0.0)

 ob1. deposit (1000);

 ob1. CalInt (1000, 5);

 ob1. displayBalance ();

 ob1. withdraw (500);

 Bank ob2 = new Bank ("Preethi", 54321, 1000.0, 50)

 ob2. deposit (2000);

 ob2. withdraw (1500);

 ob2. withdraw (500);

Deposited

Balance is : 1000.0

Antaksha V R, 1RM28CS001

Balance is : 1338.225

Deducted : 500

New Balance is : 838.225

Deposited

Money can't be withdrawn to maintain min balance

Pay servicecharge : 60.0

Balance is : 1500.0

S
17/2/24

class Bank {

} public static void main (String [] args)

Savact ob1 = new Savact ("Aketa", 193456, 0.0);

ob1. deposit (1000);

ob1. CalInt (1000, 5);

ob1. displayBalance ();

ob1. withdraw (500);

Cuacct ob2 = new Cuacct ("Sneha", 57829, 1000.0, 50);

ob2. deposit (2000);

ob2. withdraw (1500);

ob2. withdraw (500);

Deposited

Balance is : 1000.0

Akantsha V R, 1BM98CS001

Balance is : 1338.925

Deducted : 500

New Balance is : 838.925

Deposited

Money can't be withdrawn to maintain min balance

Pay servicecharge : 60.0

Balance is : 1500.0

17/12/24

C:\Users\skc\Desktop\ooj p>java Bank01

Aakanksha V R,1BM22CS001

Choose account type:

- 1. Current
- 2. Savings

Enter choice (1 or 2): 1

Enter customer name: AB

Enter account number: 5

Enter initial balance: \$1200

Enter withdrawal amount: \$500

Withdrawal successful. Current Balance: \$700.0

Account Number: 5

Customer Name: AB

Account Type: Current

Balance: \$700.0

```
C:\Users\skc\Desktop\ooj p>java Bank01
Aakanksha V R,1BM22CS001
Choose account type:
1. Current
2. Savings
Enter choice (1 or 2): 2
Enter customer name: CD
Enter account number: 6
Enter initial balance: $1500
Enter withdrawal amount: $1100
Withdrawal successful. Current Balance: $400.0
Enter interest rate: 5
Account Number: 6
Customer Name: CD
Account Type: Savings
Balance: $400.0
Enter term (in years) for compound interest calculation: 2
Compound Interest deposited. Current Balance: Rs.52900.0
Account Number: 6
Customer Name: CD
Account Type: Savings
Balance: $52900.0
```

Program - 6

- a) Create a package CTF which has two classes - Student & Internals. The class Personal has members like USN, Name, Sem. The class Internals has an array that stores the internal marks scored in 5 courses of current semesters of student. Create another package BEF which has class External derived from Student. This class has an array that stores BEF marks scored by five courses - Declare final marks of n students in all 5 courses.

```

package CTF;
import java.util.*;
public class Student
{
    public String usn;
    public String name;
    public int sem;

    public void input()
    {
        Scanner in = new Scanner (System.in);
        System.out.println ("Enter usn, name, sem: ");
        usn = in.nextLine();
        name = in.nextLine();
        sem = in.nextInt();
    }
}

```

```

package CTF;
public class Internals
{
    public int intern[] = new int [5];
}

```

```

package BEF;
import CTF.Student;
public class External extends Student

```

```
{
    public int segment[] = new int[5];
}
```

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

```
import Scanner;
```

```
import CIF;
```

```
public class Marks
```

```
{
    public static void main (String args[])
    {
        int m[] = new int[5];
    }
}
```

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

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

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

```
Scanner se[] = new Scanner [n];
```

```
Intervals si[] = new CIF [n];
```

```
for (int i=0; i<n; i++)
```

```
{ se[i] = new Scanner ();
```

```
si[i] = new CIF ();
```

```
System.out.println ("Enter details of " + (i+1));
```

```
se[i].nextLine();
```

```
for (int j=0; j<5; j++)
```

```
System.out.println ("Enter marks of " + (j+1));
```

```
se[i].nextInt();
```

```
si[i].setm[j] = in.nextInt();
```

```
in[j] = si[i].setm[j] + se[i].nextInt();
```

```
se[i].close();
```

```
System.out.println ("Final marks of " + se[i].next());
```

```
for (int k=0; k<5; k++)
```

```
{ System.out.println ("Course " + (k+1) + " = " + m[k]); }
```

```
}, }, }
```

Algorithm:

Step 1: Start

Step 2: Create a package CIE

Step 3: Declare a public class Student with members
roll, name, sem.

Step 4: A method input allows users to give inputs
for variables roll, name & sem.

Step 5: Create another class Internals in same package CIE
It consists of an array intm that stores
internal marks scored in 5 courses

Step 6: Create package SEE

Step 7: Import Student class from package CIE

Step 8: Create a class External which extends Student
class.

Step 9: It consists of an array seem that stores
external marks of 5 courses of a student.

Step 10: Import packages CIE & SEE into a
new file.

Step 12: Create an array m that stores total marks

Step 13: Read the user input for no of students

Step 14: Instantiate objects Internals & Internals

Step 15: For each student object, call input fn to
get student details.

For each course, take user input for
internal & external marks

Step 16: Add internal & external marks.

Step 17: Display total marks of all students.

Step 18: Stop.

Akanksha

IBM22C8001

Enter no of students:

1

Enter details:

Enter reg, un and name:

Enter internal and see marks of sub1

45

45

Enter internal & see marks of sub2

46

46

Enter internal & see marks of sub3

47

47

Enter internal & see marks of sub4

48

48

Enter internal & see marks of sub5

49

49

Final marks of IBM22C8001

Course1 = 90

Course2 = 92

Course3 = 94

Course4 = 96

Course5 = 98

✓
13.02.24

```
C:\Users\STUDENT\Desktop\1bm22cs029\oop>javac -d . finalMarks.java
```

```
C:\Users\STUDENT\Desktop\1bm22cs029\oop>java finalMarks
```

```
Aakanksha
```

```
1bm22cs001
```

```
enter no of students:
```

```
1
```

```
Enter details1
```

```
Enter sem,usn and name:
```

```
3
```

```
1bm22cs001
```

```
Enter internal and see marks of sub1
```

```
45
```

```
45
```

```
Enter internal and see marks of sub2
```

```
46
```

```
46
```

```
Enter internal and see marks of sub3
```

```
47
```

```
47
```

```
Enter internal and see marks of sub4
```

```
48
```

```
48
```

```
Enter internal and see marks of sub5
```

```
49
```

```
49
```

```
Final marks of 1bm22cs001
```

```
Course1=90
```

```
Course2=92
```

```
Course3=94
```

```
Course4=96
```

```
Course5=98
```

15/2/24

Program - 7

- Q) Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes age & throws the exception WrongAge() when the input age < 0. In Son class, implement a constructor that takes both father & son's age & throws an exception if son's age = father's age.

Algorithm

1. Start
2. Create an user defined exception WrongAge that throws an error.
3. Create a base class Father.
4. A constructor is used accept the father's age and if age < 0, throws an exception WrongAge that prints Age cannot be negative.
5. Create a class Son that extends to another class Father.
6. Create a constructor to accept sonage & get data of father's age from superclass.
7. If throws an exception WrongAge if son's age is greater than father's age.
8. Create a class FatherSon & take input for father's & son's age.
9. Write a try and catch block to get the find the error.
10. Stop Create an object s of Son datatype and pass sonage & father age as arguments.
11. Print son's age
12. Print father's age
13. If catch block encounters any exception e, it print error
14. Stop

```
import java.util.Scanner;  
class WrongAge extends Exception {  
    public WrongAge(String message) {  
        super(message);  
    }  
}
```

```
class Father {  
    int fatherAge;  
    public Father(int fatherAge) throws  
        WrongAge {  
        if (fatherAge < 0) {  
            throw new WrongAge("Age cannot be negative");  
        this.fatherAge = fatherAge;  
    }  
}
```

```
class Son extends Father {  
    int sonAge;  
    public Son(int fatherAge, int sonAge)  
        throws WrongAge {  
        super(fatherAge);  
        if (sonAge >= fatherAge) {  
            throw new WrongAge("Son's age must be less than  
                father's age");  
        this.sonAge = sonAge;  
    }  
}
```

```

public class FatherSon {
    public static void main (String [] args) {
        Scanner in = new Scanner (System.in);
        System.out.println ("Enter father's age & son's age");
        int fage = in.nextInt();
        int sage = in.nextInt();
        try {
            Son s = new Son (fage, sage);
            System.out.println ("Father's age: " + s.fatherAge);
            System.out.println ("Son's age: " + s.sonAge);
        } catch (WrongAge e) {
            System.out.println ("Error: " + e.getMessage());
        } catch (Exception e) {
            System.out.println ("Error: " + e);
        }
    }
}

```

Q1. Enter father's age & son's age

45

32

Father's age: 45

Son's age: 32

Aakanksha V.R., IIBM22C8061

Q2. Enter father's age & son's age

-8

92

ERROR: Age cannot be negative

Q3. Enter father's age & son's age

22

25

ERROR: Son's age must be less than father's age

```
C:\Users\bmsce\Desktop\1BM22CS001>java fatherson
Enter father's age and son's age
45
32
Father's age:45
Son's age :32
Aakanksha V R,1BM22CS001
```

```
C:\Users\bmsce\Desktop\1BM22CS001>java fatherson
Enter father's age and son's age
-8
22
ERROR:Age cannot be negative
Aakanksha V R,1BM22CS001
```

```
C:\Users\bmsce\Desktop\1BM22CS001>java fatherson
Enter father's age and son's age
23
25
ERROR:Son's age must be less than father's age
Aakanksha V R,1BM22CS001
```

Program 8

Date 15/2/24
Page

- Write a program which creates 2 threads, one thread displaying "BMS College of Engineering" once every ten seconds & another displaying "CSE" once every two seconds.

Algorithm

1. Start
2. Create a class A that extends to Thread.
3. Initialize $f1 = 0$, time = 30000 in constructor of A
4. Write a ~~method~~ run.
5. Use a while loop which runs until the condition $f1 <= \text{time}$ becomes false.
6. Print BMS College of Engineering.
Include try, block and use sleep method for 10000s.
In catch block, print the statement error if it encounters exception e. Increment $f1 = f1 + 15000$;
7. Create a class B that extends to Thread.
8. Initialize $f2 = 0$, time = 30000 in constructor of B.
9. Write a method run.
Use a while loop which runs until the condition $f2 <= \text{time}$ becomes false.
Print CSE
Include try block and call sleep method for 2000s.
In catch block, print the statement error if it encounters exception e.
Increment $f2 = f2 + 2000$;
10. Create a class Demo.
11. Instantiate object a of datatype A, b of datatype B.
12. Call method start of object A
13. Call method start of object B.
14. Stop.

class A extends Thread

{ int f1, time;

A() {

f1 = 10000;

time = 30000;

}

public void run()

{ while (f1 <= time)

{ System.out.println ("SMB (College of Engineering);

try

{ sleep (10000);

}

catch (InterruptedException e) {

System.out.println ("error");

}

f1 += 10000;

}

}

int f2, time;

B() {

time = 30000;

f2 = 5000;

}

public void run()

{ while (f2 <= time)

{ System.out.println ("CSE");

try {

sleep (5000);

}

catch (Exception e)
{ System.out.println ("error"); }

$$t2 + = 2000;$$

23

class Demo

```
{ public static void main (String args[])
{ A a = new A();
B b = new B();
a.start();
b.start(); }}
```

O/P: Ankusha.VR , IBM22G1001

BMS COLLEGE OF ENGINEERING

CSE

CSE

CSE

CSE

CSE

BMS COLLEGE OF ENGINEERING

CSE

CSE

CSE

CSE

CSE

16.09.2024
BMS COLLEGE OF ENGINEERING
CSE
CSE
CSE

C:\Users\bmsce\Desktop\1BM22CS001>java program8demo

Aakanksha V R, 1BM22CS001

CSE

BMS COLLEGE OF ENGINEERING

CSE

CSE

CSE

CSE

BMS COLLEGE OF ENGINEERING

CSE

CSE

CSE

CSE

CSE

BMS COLLEGE OF ENGINEERING

CSE

CSE

CSE

Program 9

Q)

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 & Num2. The division of Num1 or Num2 here not an Integer, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
```

```
class SwingDemo {
    SwingDemo() {
        JFrame frm = new JFrame("Divide App");
        frm.setSize(275, 150);
        frm.setLayout(new FlowLayout());
        frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JLabel glab = new JLabel("Enter dividend & divisor:");
        JTextField dgtf = new JTextField(8);
        JTextField bgtf = new JTextField(8);
        JButton button = new JButton("Calculate");
    }
}
```

```
JLabel err = new JLabel();
JLabel dlab = new JLabel();
JLabel blab = new JLabel();
JLabel anslab = new JLabel();
```

```
frm.add(err);
frm.add(dlab);
frm.add(blab);
frm.add(anslab);
frm.add(button);
```

frm.add(abb);
 frm.add(bbb);
 frm.add(anslab);

ActionListener I - new ActionListener() {

public void actionPerformed(ActionEvent evt) {
 System.out.println("Action event from text field");
 }

}

ajtf.addActionListener(I);
 bftf.addActionListener(I);

button.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent evt) {
 System.out.println("Action event from a text
 field");
 }

ajtf.addActionListener(I);

bftf.addActionListener(I);

button.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent evt)

{ try

int a = Integer.parseInt(ajtf.getText());

int b = Integer.parseInt(bftf.getText());

int ans = a/b;

alab.setText("In A=" + a);

blab.setText("In B=" + b);

anslab.setText("In Ans=" + ans);
 }

catch (NumberFormatException e)

{ alab.setText("In");

blab.setText("In");

anslab.setText("In");

```
err. setText ("Enter only integers!"); }
```

```
catch (ArithmaticException e) {
```

```
ab.setError ("");
```

```
blab.setText ("");
```

```
anlab.setText ("");
```

```
err.setText ("B should be NON zero!");
```

```
}
```

```
});
```

```
frm.setVisible (true);
```

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

```
SwingUtilities.invokeLater (new Runnable()) {
```

```
public void run() {
```

```
new SwingDemo ();
```

```
};
```

```
}
```

Q1:

OP1

Dividend

95

Divisor

4

$a=95, b=4, d_{AB} = 6$

OP2

Dividend

25

Divisor

0

B should be NON zero!

OP3

Dividend

25.4

Divisor

4

Enter only integers!

actionPerfomed → This method is invoked automatically whenever you click on the registered component.

ActionListener :- This is notified whenever you click on the button or menu item. It is notified against ActionEvent. ActionListener interface is found in `java.awt.event` package. It has only one method ; `actionPerformed()`.

setText method substitutes the characters for the text in text field.

getText method returns the text from single-line text field.

setVisible method makes the frame appear on the screen.

invokeLater() method is a static method of SwingUtilities class and it can be used to perform a task asynchronously in the AWT Event dispatcher thread.

JLabel is a built -in Java Swing class that lets you display information on a JFrame. This element cannot be changed by the user. However it can be changed by other program statements.

23.02.2014

B should ne NON zero! Enter the dividend and divisor: Aakanksha V R,1BM22CS001

Enter only integers! Enter the dividend and divisor: Aakanksha V R,1BM22CS001 A=24 B=5 Ans=4

Enter only Integers! Enter the dividend and divisor: Aakanksha V R,1BM22CS001