

i) Java program to print an integer Entered by the user

. Algorithm:

Step 1: Start

Step 2 : Take input from user by creating reader

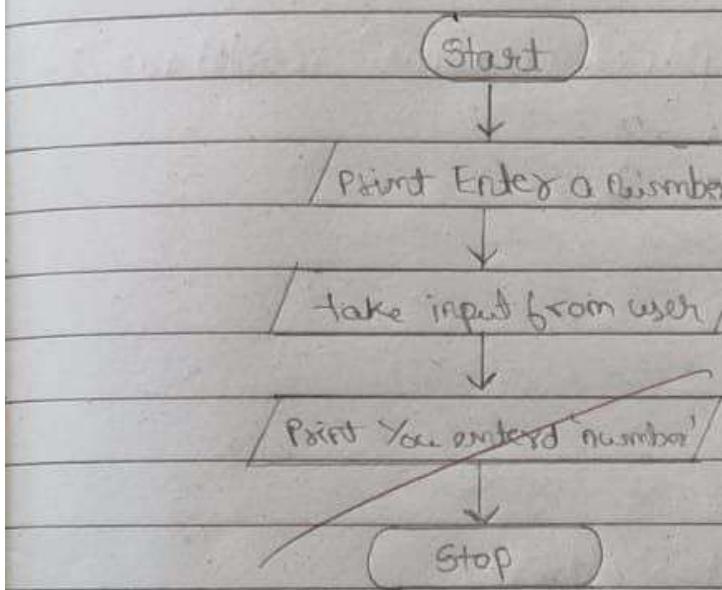
Step 3: Enter a number (print) instance

Step 4: Read a number entered by user

Step 5: Print the number entered by user

Step 6 : Stop

. Flowchart:



• Code

```
import java.util.Scanner;  
public class HelloWorld {  
    public static void main (String [] args)
```

Scanner sc = new Scanner (System.in);
System.out.println ("Enter a number: ");
int number = sc.nextInt();
System.out.println ("You entered: " + number);

• Output

Enter a number : -2

You entered : -2

```
C:\Users\bmsce>CD C:\Users\bmsce\Desktop\1bm22cs027
C:\Users\bmsce\Desktop\1bm22cs027>javac HelloWorld.java
C:\Users\bmsce\Desktop\1bm22cs027>java HelloWorld
Akanksha Singa
1BM22CS027
Enter a number: -2
You entered: -2
```

2) Java program to check whether a number is even or odd

• Algorithm

Step 1 : Start

Step 2 : Print Enter an integer number

Step 3 : Take input from the user and store it in a variable num

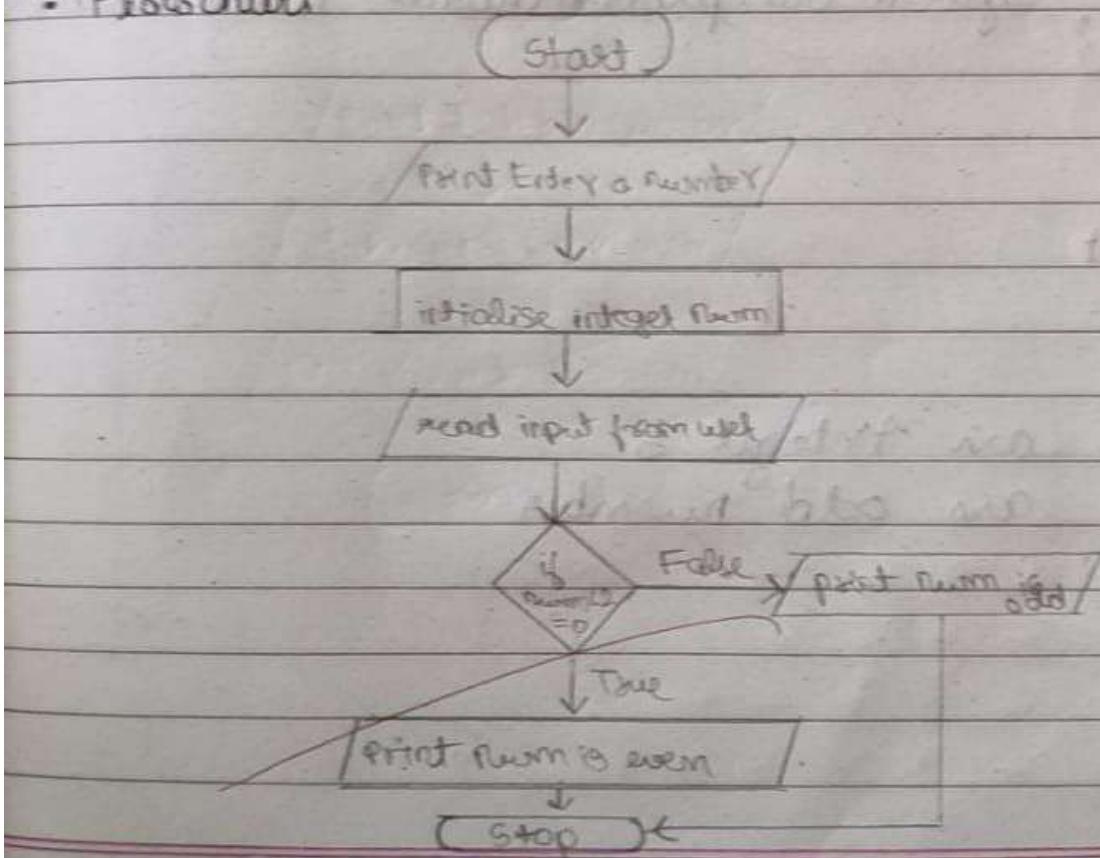
Step 4 : Check if $\text{num} \% 2$ gives remainder 0

Step 5 : If true print number is an even no and go to step 7 else go to step 6

Step 6 : print number is an odd no

Step 7 : Stop

• Flowchart



• Program

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

```
public class JavaExample
```

```
{
```

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

```
    {
```

```
        int num;
```

```
        System.out.print("Enter an integer number:");
```

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

```
        num = sc.nextInt();
```

```
        if (num % 2 == 0)
```

```
            System.out.println(num + " is an even number");
```

```
        else
```

```
            System.out.println(num + " is odd number");
```

```
}
```

```
}
```

• Output

Enter an integer: 5

5 is an odd number

```
C:\Users\bmsce\Desktop\1bm22cs027>javac JavaExample.java
C:\Users\bmsce\Desktop\1bm22cs027>java JavaExample
Akanksha Singa
1BM22CS027
Enter an Integer: 5
5 is an odd number
```

Q) Java Program to print right triangle star pattern with 8 no of rows.

Algorithm

Step 1 : Start

Step 2 : Initialise rows, column, no of rows = 8

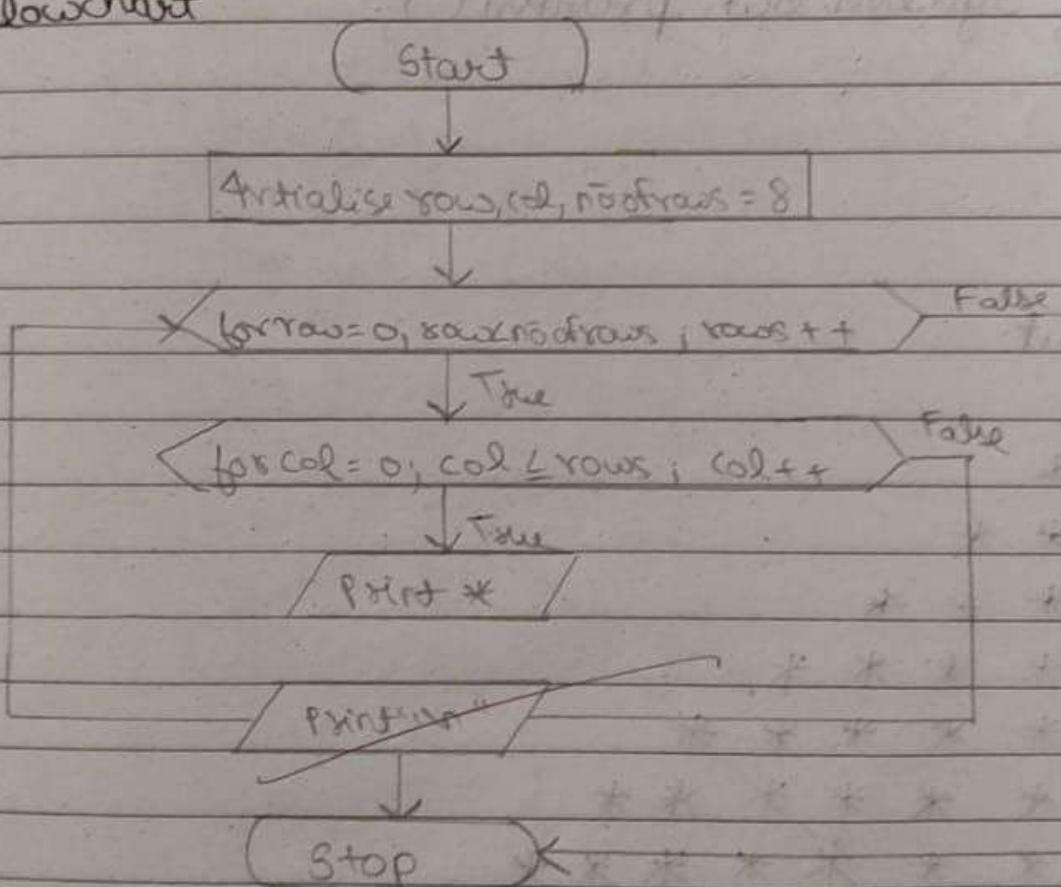
Step 3 : for rows less than no of rows increase rows by one else goto step 6

Step 4 : for columns less than no of rows increase column by one else goto Step 3 & print star

Step 5 : print *

Step 6 : Stop

Flowchart



• Program:

```
public class Triangle;
```

```
{
```

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

```
{
```

```
        int row, column, numberofrows = 8;
```

```
        for (row = 0; row < numberofrows; row++)
```

```
{
```

```
            for (column = 0; column < numberofrows; column++)
```

```
{
```

```
                System.out.print ("*");
```

```
}
```

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

```
}
```

```
3
```

• Output

```
*
```

```
* *
```

```
* * *
```

```
* * * *
```

```
* * * * *
```

```
* * * * *
```

```
* * * * *
```

```
* * * * *
```

```
C:\Users\bmsce\Desktop\1bm22cs027>javac Triangle.java

C:\Users\bmsce\Desktop\1bm22cs027>java Triangle
Akanksha Singa
1BM22CS027
*
**
***
****
*****
******
*****
```

4) Java program to Find Quotient and Remainder of 15 and 2

- Algorithm

- Step 1 : Start

- Step 2 : Initialise num1 = 15 and num2 = 2

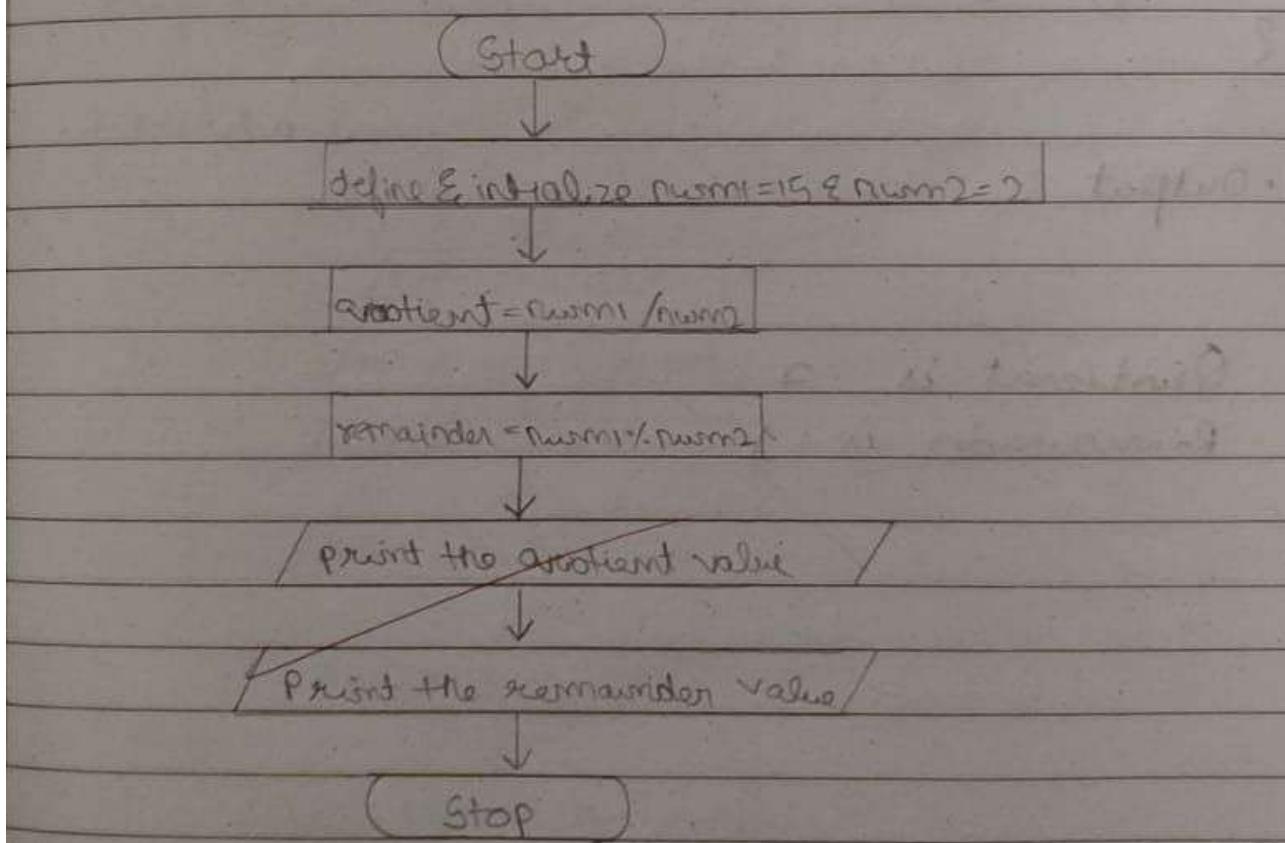
- Step 3 : Initialise quotient = num1 / num2

- Step 4 : Initialise remainder = num1 % num2

- Step 5 : Print values of quotient & remainder

- Step 6 : Stop

- Flowchart



• Program:

```
import java.util.Scanner;  
Public class QandR  
{  
    public static void main (String [] args)  
    {
```

Scanner

```
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);
```

3

• Output

Quotient is : 7

Remainder is : 1

```
C:\Users\bmsce\Desktop\1bm22cs027>javac QandR.java  
C:\Users\bmsce\Desktop\1bm22cs027>java QandR  
Akanksha Singa  
1BM22CS027  
Quotient is : 7  
Remainder is: 1
```

5) Java Program to Multiply Two Numbers

• Algorithm

Step 1 : Start

Step 2 : Print Enter first number

Step 3 : Initialise num1 of type int and read input from user

Step 4 : Print Enter second number

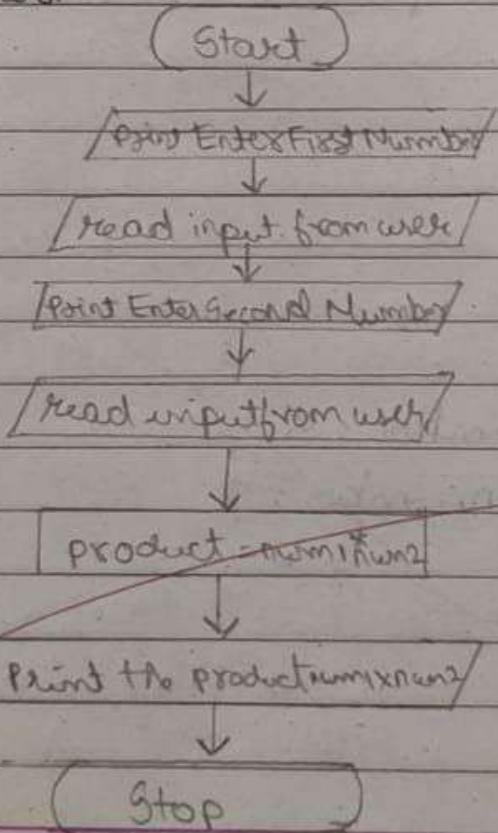
Step 5 : Initialise num2 of type int and read input from user

Step 6 : Calculate product = num1 * num2

Step 7 : Print product

Step 8 : Stop

• Flowchart



- Program

```
Import java.util.Scanner;
```

```
public class Multiply
```

```
{
```

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

```
{
```

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

```
    System.out.print ("Enter first number: ");
```

```
    int num1 = sc.nextInt();
```

```
    System.out.print ("Enter second number: ");
```

```
    int num2 = sc.nextInt();
```

```
    sc.close();
```

```
    int product = num1 * num2;
```

```
    System.out.println ("num1 * num2 = " + product);
```

3

3

- Output

Enter first number: 5

Enter second number: 2

$5 \times 2 = 10$

```
C:\Users\bmsce\Desktop\1bm22cs027>javac Multiply.java
```

```
C:\Users\bmsce\Desktop\1bm22cs027>java Multiply
```

```
Akanksha Singa
```

```
1BM22CS027
```

```
Enter first number: 5
```

```
Enter second number: 2
```

```
5 x 2 = 10
```

Q) Swap two floating point numbers 1.2 & 2.45

• Algorithm

Step 1 : Start

Step 2 : Define first and second as float datatype
and initialize to 1.2 & 2.45

Step 3 : Print statement before swap

Step 4 : Print first and second number

Step 5 : Create a temp variable of float datatype
and store value of first variable

Step 6 : Store variable second in first

Step 7 : Store temp in second variable

Step 8 : Print statement after swap

Step 9 : Print first and second number

Step 10 : Stop

(Start) .

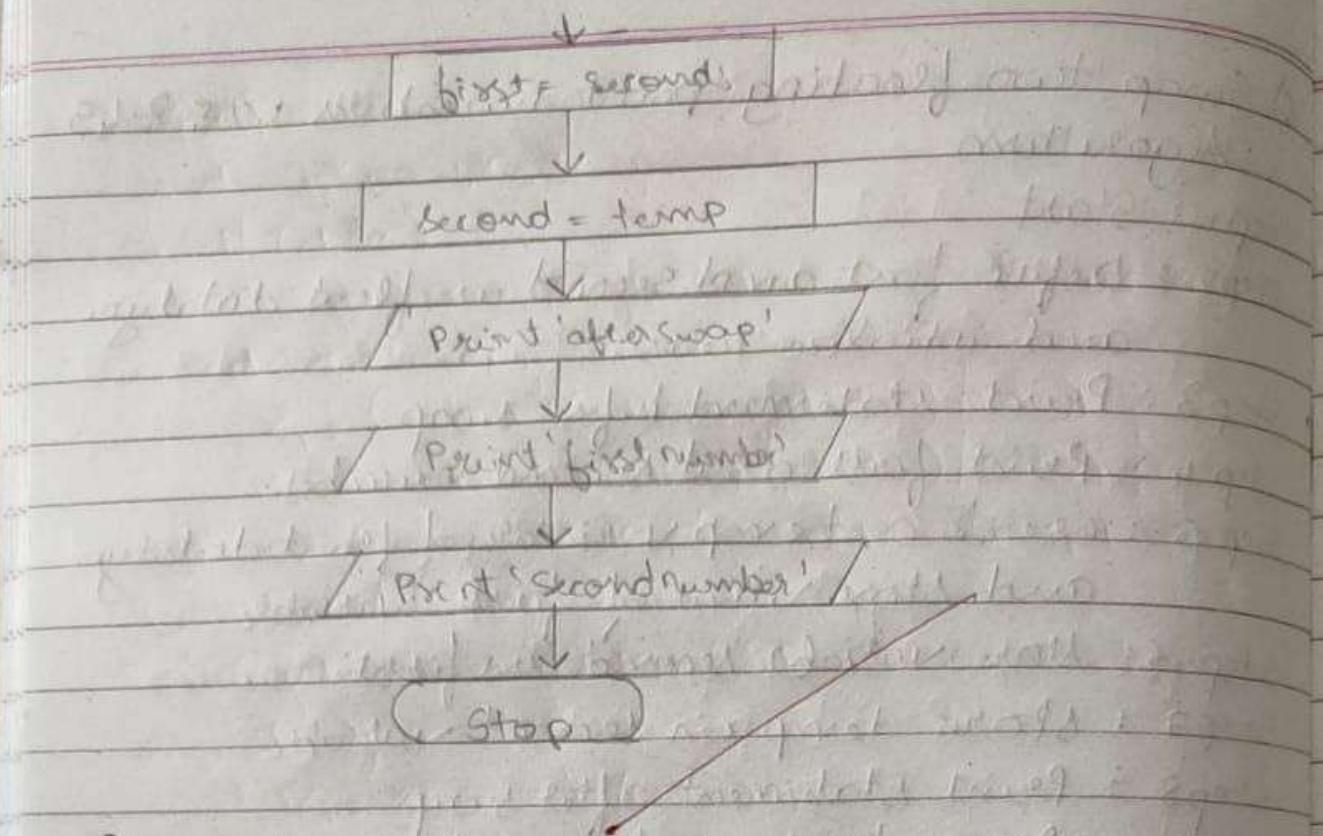
↓
Define and initialize first = 1.2f & second = 2.4f

↓
Print 'Before Swap'

↓
Print 'first number'

↓
Print 'second number'

↓
temp = first



- Program

```
public class Swap {
```

```
    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);
```

```
}
```

```
3
```

- Output

-- Before Swap --

First Number = 1.2

Second Number = 2.45

-- After Swap --

First Number = 2.45

Second number = 1.2

```
C:\Users\bmsce\Desktop\1bm22cs027>javac Swap.java  
  
C:\Users\bmsce\Desktop\1bm22cs027>java Swap  
Akanksha Singa  
1BM22CS027  
--Before Swap--  
First number= 1.2  
Second number= 2.45  
--After Swap--  
First number= 2.45  
Second number= 1.2
```

→ WAP that prints all real solutions to the quadratic equation $an^2 + bn + c = 0$. Read in a, b, c and use the quadratic formula. If the discriminant $b^2 - 4ac$ 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 Quadratic {
    public static void main (String [] args) {
        float a, b, c, d;
        double r1, r2;
        Scanner sc = new Scanner (System.in);
        System.out.print ("Enter coefficients.");
        a = sc.nextFloat ();
        b = sc.nextFloat ();
        c = sc.nextFloat ();
        if (a == 0)
            System.out.println ("Invalid input");
```

else {

$$d = b * b - 4 * a * c;$$

if ($d > 0$)

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

$$\gamma_1 = (-b + \text{Math.sqrt}(d)) / (2 * a);$$

$$\gamma_2 = (-b - \text{Math.sqrt}(d)) / (2 * a);$$

System.out.println ("root1 = " + γ_1 + "root2 = "
} else if ($d == 0$)

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

$$\gamma_1 = \gamma_2 = -b / (2 * a);$$

System.out.println ("root1 = " + γ_1 + "root2 = "
}

else {

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

$$\gamma_1 = -b / (2 * a);$$

$$\gamma_2 = \frac{\text{Math.sqrt}(\text{abs}(d))}{(2 * a)};$$

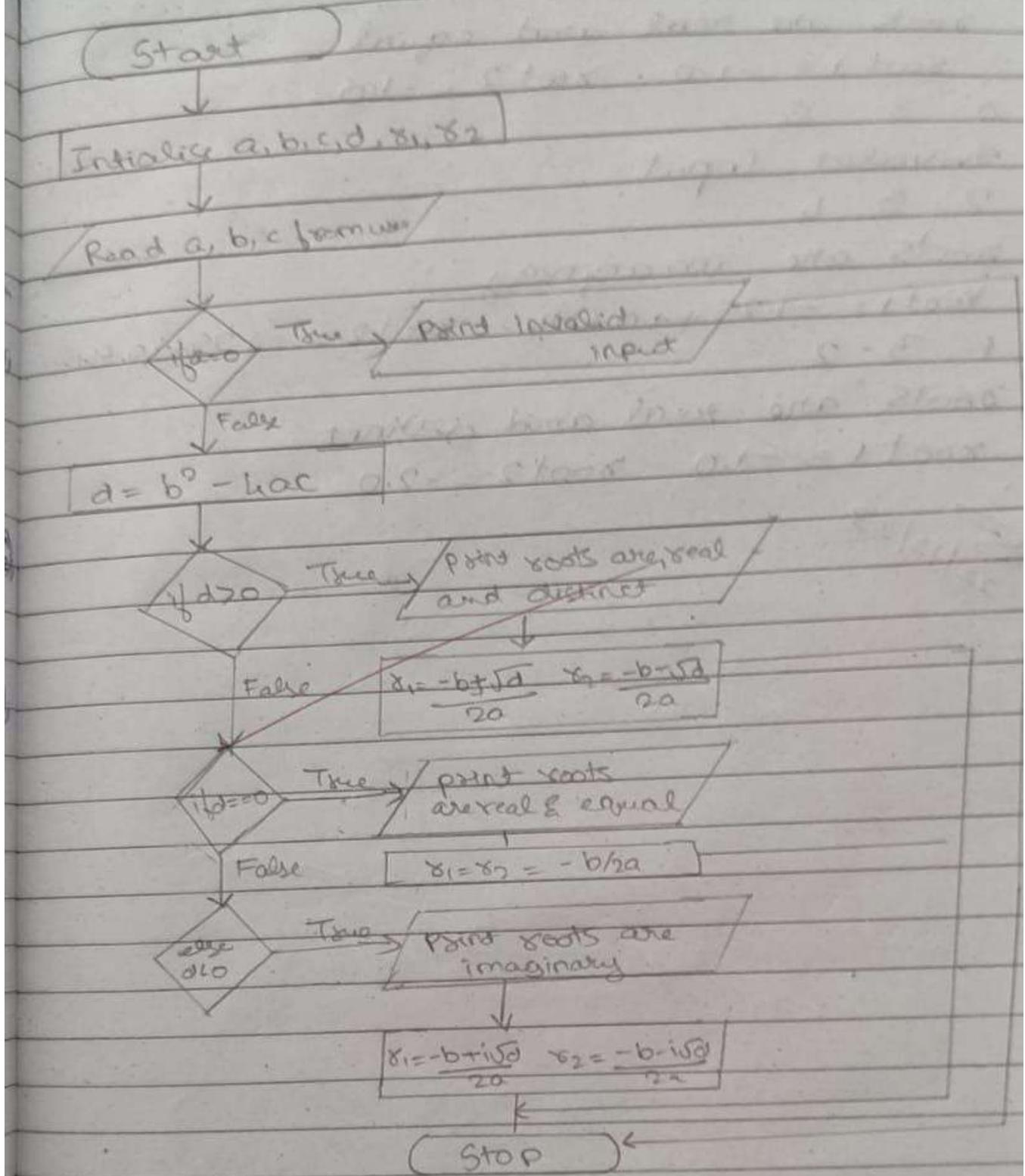
System.out.println ("root1 = " + γ_1 + "i" + γ_2 +
"root2 = " + γ_2 + "-i" + γ_1);

}

}

}

Flow chart



Output:

2 4 2

roots are real and equal

$$\text{root 1} = -1.0 \quad \text{root 2} = -1.0$$

0 3 5

Invalid input

2 5 4

roots are imaginary

$$\text{root 1} = -1.25 + i0.6614 \quad \text{root 2} = -1.25 - i0.6614$$

1 3 2

roots are real and distinct

~~$$\text{root 1} = -1.0 \quad \text{root 2} = -2.0$$~~

~~22/12/23~~

3.

```
C:\Users\bmsce\Desktop\1bm22cs027 ooj>javac Quadratic.java

C:\Users\bmsce\Desktop\1bm22cs027 ooj>java Quadratic
AKANKSHA SINGA
1BM22CS027
2
4
2
roots are real and equal
root1= -1.0  root2= -1.0

C:\Users\bmsce\Desktop\1bm22cs027 ooj>javac Quadratic.java

C:\Users\bmsce\Desktop\1bm22cs027 ooj>java Quadratic
AKANKSHA SINGA
1BM22CS027
0
3
5
Invalid input

C:\Users\bmsce\Desktop\1bm22cs027 ooj>javac Quadratic.java

C:\Users\bmsce\Desktop\1bm22cs027 ooj>java Quadratic
AKANKSHA SINGA
1BM22CS027
2
5
4
roots are imaginary
root1= -1.25+ i0.6614378277661477  root2= -1.25- i0.6614378277661477

C:\Users\bmsce\Desktop\1bm22cs027 ooj>javac Quadratic.java

C:\Users\bmsce\Desktop\1bm22cs027 ooj>java Quadratic
AKANKSHA SINGA
1BM22CS027
1
3
2
roots are real and distinct
root1= -1.0  root2= -2.0
```

8) WAP to calculate the SGPA of a given person and display its details

→ import java.util.Scanner
public class Student

2

```
String name;  
String USN;  
int credits[8] = {4, 4, 4, 3, 3, 2, 1};  
int marks[] = new int[8];  
Scanner sc = new Scanner(System.in);  
public void details()
```

3

```
System.out.print("Enter your  
name");  
name = sc.next();  
System.out.print("Enter your USN")  
USN = sc.next();
```

3

```
public void marks()
```

```
System.out.print("Enter your  
marks in correct order");
```

~~for~~

```
System.out.print("Enter 10 for 90-100  
9 for 80-89, 8 for 70-79 and so on");  
for (int i=0; i<8; i++)
```

~~temp += credits[i] * marks[i]~~

```
5 marks[i] = sc.nextInt();  
public double sgpa()  
5 double sgpa, temp;  
for (i=0; i<8; i++)  
    if (marks[i]>100)  
        temp += Credits[i]*marks[i];  
    else  
        sgpa = temp / 22;  
3 return sgpa;
```

```
public void display()
```

```
5 System.out.print("Name: " + name);  
System.out.print("USN: " + usn);  
System.out.print("SGPA: " + sgpa);
```

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

```
5 Student s1 = new Student();  
s1.details();  
s1.marks();  
s1.display();
```

```
3  
Output: Enter your name: Akanksha  
Enter your USN: IBM22CG027
```

```
100 95 90 92 89 90 85 93
```

```
Name: Akanksha
```

```
USN: IBM22CG027
```

Algorithm:

Start:

- 1) Initialise variables of Name, USN of type string and credits and marks as int type of array
- 2) Initialise values of credits as credits $\{ \} = \{ 4, 4, 4, 3, 3, 2, 1 \}$
- 3) Take user input for name, usn and marks and store it
- 4) calculate Sgpa as credits * marks divided by 22
- 5) display details using display function
- 6) ~~create a variable s1 of type student and access different methods with it.~~
- 7) Stop

Fix
29/12/23

```
C:\Users\bmsce\Desktop\1bm22cs027 ooj>cd C:\Users\bmsce\Desktop\1bm22cs027 ooj
C:\Users\bmsce\Desktop\1bm22cs027 ooj>javac Student.java
C:\Users\bmsce\Desktop\1bm22cs027 ooj>java Student
AKANKSHA SINGA
1BM22CS027
Enter your name:
Akanksha
Enter your usn:
1BM22CS027
Enter your marks in same order:
100
95
90
92
89
90
85
93
Name: Akanksha
USN:1BM22CS027
SGPA:9.818181818181818

C:\Users\bmsce\Desktop\1bm22cs027 ooj>
```

• Develop a java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide classes Rectangle, Triangle & Circle that extends Shape abstract class Shape

{

int i, j;

public abstract void printArea();

}

Class Rectangle extends Shape

{

Rectangle (int i, int j)

{

this.i = i;

this.j = j;

}

public void printArea()

{

System.out.println ("area of rectangle")

= " + i * j);

}

}

Class Triangle extends Shape

{

Triangle (int i, int j)

{

this.i = i;

abstract
integers
provide
shape

3 this.j = j;

{

public void printArea()

{

System.out.println("area of Triangle:
" + i*j/2);

{

3

class Circle extends Shape

{

Circle (int i)

{

this.i = i;

{

public void printArea()

{

System.out.println("area of circle:
" + 3.14 * i * i);

{

3

class Area

{

public static void main (String args[])

{

Rectangle r = new Rectangle (10, 20);

Triangle t = new Triangle (10, 20);

Circle c = new Circle (5);

x. paintArea();
t. paintArea();
c. paintArea();

3

3

Algorithm

- 1) Create a abstract class called Shape
- 2) Create an abstract method `paintArea` which you can't create an instance
- 3) To access `paintArea` method as it is object create subclass
- 4) Create subclass Rectangle, Circle, Triangle and Constructors which takes values of user input and evaluate it
- 5) Create a main function in same or different class
- 6) Create instance objects for the class rectangle, circle and triangle and pass parameters to function.

Stop

Output:

Area of rectangle: 200

Area of triangle: 100

Area of circle: 78.5

```
C:\Users\bmsce\Desktop\1bm22cs027 ooj>javac Area.java  
C:\Users\bmsce\Desktop\1bm22cs027 ooj>java Area  
AKANKSHA SINGA  
1BM22CS027  
area of Rectangle: 200  
area of Triangle: 100  
area of Circle: 78.5
```

• Create a class Book which contains four members, name, author, price num-pages. Include a constructor to set the values for the members. Include methods to set and get the details of the object. Include toString() method that could display the complete details of the book. Develop a java program to create n book objects.

→ import java.util.Scanner;
class Book.

String name;
String author;
float price;
int num-pages;

3

Void setdetails()

Scanner sc = new Scanner(System.in);
System.out.print("enter your
name:");
name String = sc.next();
System.out.print("enter author name")
String author = sc.next();
System.out.print("enter price of
book:");

```
    price = sc.nextDouble();  
    System.out.print(" enter number of  
    pages in book");
```

```
    numPages = sc.nextInt();
```

```
void setDetails
```

```
    toString();
```

```
void toString()
```

```
System.out.println(" Book details : ");
```

```
Name of Book : the name + " Author : " + author  
+ " Price : " + price + " No of pages " + numPages
```

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

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

```
System.out.println (" Enter no of books ")
```

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

```
int b[] Book b[] = new Book[n];
```

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

```
b[i] = new Book();
```

```
b[i]. setDetails();
```

```
System.out.println (" book details ");
```

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

b[i]. get details;

3

3 enter no of books : 3

enter book name : abc

author name : def

enter price : 200

enter no of pages : 100

enter book name : xyz

enter author name :uvw

enter price : 100

enter no of pages : 50

details of book : name : abc author : def

price : 200 num-pages : 100

details of book : name : xyz author : uvw

Price 100 num-pages: 50

Algorithm :

- 1) Start
- 2) Create a class Book which has variable that describes details of the Book (name, author, price, num-pages)
- 3) Create a function setDetails which takes input from user for details of each book through main function
- 4) Create a function getDetails which calls the toString method to describe all the details of a given book
- 5) Create a toString method which prints the details of a particular Book
- 6) Create a main method initialise an array b of type Book and create instance of it at each index
- 7) Stop

~~Final Draft
of See~~

```
C:\Users\bmsce>cd C:\Users\bmsce\Desktop\1bm22cs027 ooj  
C:\Users\bmsce\Desktop\1bm22cs027 ooj>javac Book.java  
C:\Users\bmsce\Desktop\1bm22cs027 ooj>java Book  
AKANKSHA SINGA  
1BM22CS027  
Enter no.of books:  
3  
enter book name:  
abc  
enter author name:  
def  
enter price:  
200  
enter no. of pages:  
100  
enter book name:  
xyz  
enter author name:  
uvw  
enter price:  
100  
enter no. of pages:  
50  
enter book name:  
fgh  
enter author name:  
hij  
enter price:  
300  
enter no. of pages:  
500  
details of book: name:abcauthor: defprice:200.0num_pages:100  
details of book: name:xyzauthor: uvwprice:100.0num_pages:50  
details of book: name:fghauthor: hijprice:300.0num_pages:500
```

Write a Java program to create a class Bank that maintains two kinds of account for its customer. One called savings account and the other current account. The saving account provides CI and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the classes cur-acct and sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

- 1) Accept deposit from customer & update the balance
- 2) Display the balance
- 3) Compute & deposit interest
- 4) Permit withdrawal and update the balance. Check min balance imposed

import java.util.Scanner;
class Account

String customerName;
long accno;
String accountType;
double balance;
public Account (String customerName,
long accno, String accountType)

this.customerName = customerName;
this.accno = accno;
this.accountType = accountType;
this.balance = 0.0;

}

public void displayBalance ()

System.out.println ("Account Number:
" + accno);
System.out.println ("Customer Name:
" + customerName);

System.out.println ("Account Type:
" + accountType);

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

class CurrentAccount {

 double minBalance;

 double serviceCharge;

 public CurrentAccount (String customerName,
 long accno)

 { minBalance = 500.0; }

 this.minBalance = minBalance;

 this.serviceCharge = 50.0;

}

 public void withdraw (double amount)

 { if (balance - amount) >= minBalance
 { balance -= amount; }

 System.out.println ("Withdrawal
 successful. Current Balance : \$ " + balance);

 }

 else

 System.out.println ("Insufficient funds
 Withdrawal not allowed.");

}

 public void imposeServiceCharge ()

 {

 if (balance < minBalance)

 balance -= serviceCharge;

 System.out.println ("Service charge

composed. Current Balance: Rs." + balance);
}

}

class SavAcct extends Account {

double interestRate;

public SavAcct (String customerName,
long accno)

{ super (customerName, accno, "Savings");
this.interestRate = 0.05;

}

public void compoundInterest (double
initialAmount, int term)

{ double compoundInterest = initialAmount
* Math. pow ((1 + interestRate), term) - initial
Amount;

balance += compoundInterest;

System.out.println ("compound Interest
deposited. Current Balance: Rs." + balance);
}

ublic class Bank

{ public static void main (String args) {

Scanner sc = new Scanner (System.in);

System.out.println ("Choose account type:

System.out.println ("1. Current 2. Savings")

System.out.print("Enter your choice.");
int choice = sc.nextInt();

System.out.print("Enter customer name.");
String customerName = sc.next();

System.out.print("Enter account number.");
long accno = sc.nextLong();

if (choice == 1)
{
 Current curAccount = new Current
(customerName, accno);

System.out.print("Enter initial balance. \$");

double initialBalance = sc.nextDouble();

curAccount.Balance = initialBalance;

System.out.print("Enter withdrawal amount. \$");

double withdrawalAmount = sc.nextDouble();

curAccount.withdraw(withdrawalAmount);

curAccount.imposeServiceCharge();

curAccount.displayBalance();

3

else if (choice == 2)

SavAcct savAccount = new SavAcct(customerName, accno);

System.out.print("Enter initial balance. \$");

```
double initialBalance = sc.nextDouble();
savAccount.balance = initialBalance;
System.out.print("Enter withdrawal amount: $");
double withdrawalAmount = sc.nextDouble();
savAccount.balance -= withdrawalAmount;
System.out.println("Withdrawal Successful");
System.out.println("Current Balance: $" + savAccount.balance);
System.out.print("Enter interest rate: ");
double interestRate = sc.nextDouble();
savAccount.displayBalance();
System.out.print("Enter term (years) for CI calculation: ");
int term = sc.nextInt();
savAccount.compoundInterest(initialBalance, term);
savAccount.displayBalance();
} else
{
    System.out.println("Invalid choice");
}
```

Algorithm

Step 1: Start

Step 2: Initialise variables cust-name, acc-type, balance.

Step 3: Input : enter customer name , accno
balance account type as savings
or current from user

Step 4: enter the choice of savings & current
account

Step 5: Read the choice if choice is
savings then steps for current

Step 6:

Step 6: enter initial balance, withdrawal
amount & min balance from user
check condtn for min balance i.e initial-
balance - withdrawal should be \geq min
i.e 500 then New Balance = balance -
(withdrawal)

then print currentBalance enter
interest rate & time to calculate CI
$$CI = \text{balance} * \text{power}(1 + \text{interest rate}, \text{time})$$

- initial balance

Print CI & deposit with interest

Step 7: else if (choice=2)

create obj of SavAct initialize

initial balance, withdrawal amt, interest rate

Step 8: Stop

19/11
SavAct

```
C:\Users\bmsce\Desktop\1bm22cs027 ooj>javac Bank.java
```

```
C:\Users\bmsce\Desktop\1bm22cs027 ooj>java Bank
```

```
AKANKSHA SINGA
```

```
1BM22CS027
```

```
Choose account type:
```

```
1. Current
```

```
2. Savings
```

```
Enter choice (1 or 2): 1
```

```
Enter customer name: Akanksha
```

```
Enter account number: 027027027
```

```
Enter initial balance: $10000
```

```
Enter withdrawal amount: $5000
```

```
Withdrawal successful. Current Balance: $5000.0
```

```
Account Number: 27027027
```

```
Customer Name: Akanksha
```

```
Account Type: Current
```

```
Balance: $5000.0
```

```
C:\Users\bmsce\Desktop\1bm22cs027 ooj>javac Bank.java
```

```
C:\Users\bmsce\Desktop\1bm22cs027 ooj>java Bank
```

```
AKANKSHA SINGA
```

```
1BM22CS027
```

```
Choose account type:
```

```
1. Current
```

```
2. Savings
```

```
Enter choice (1 or 2): 2
```

```
Enter customer name: Akanksha
```

```
Enter account number: 027027027
```

```
Enter initial balance: $5000
```

```
Enter withdrawal amount: $500
```

```
Withdrawal successful. Current Balance: $4500.0
```

```
Enter interest rate: 0.05
```

```
Account Number: 27027027
```

```
Customer Name: Akanksha
```

```
Account Type: Savings
```

```
Balance: $4500.0
```

```
Enter term (in years) for compound interest calculation: 2
```

```
Compound Interest deposited. Current Balance: Rs.5012.5
```

```
Account Number: 27027027
```

```
Customer Name: Akanksha
```

```
Account Type: Savings
```

```
Balance: $5012.5
```

```
C:\Users\bmsce\Desktop\1bm22cs027 ooj>javac Bank.java
```

```
C:\Users\bmsce\Desktop\1bm22cs027 ooj>java Bank
```

AKANKSHA SINGA

1BM22CS027

Choose account type:

1. Current

2. Savings

Enter choice (1 or 2): 1

Enter customer name: Akanksha

Enter account number: 027027027

Enter initial balance: \$800

Enter withdrawal amount: \$400

Insufficient funds. Withdrawal not allowed.

Account Number: 27027027

Customer Name: Akanksha

Account Type: Current

Balance: \$800.0

```
C:\Users\bmsce\Desktop\1bm22cs027 ooj>javac Bank.java
```

```
C:\Users\bmsce\Desktop\1bm22cs027 ooj>java Bank
```

AKANKSHA SINGA

1BM22CS027

Choose account type:

1. Current

2. Savings

Enter choice (1 or 2): 1

Enter customer name: Akanksha

Enter account number: 027027027

Enter initial balance: \$400

Enter withdrawal amount: \$100

Insufficient funds. Withdrawal not allowed.

Service charge imposed. Current Balance: Rs.350.0

Account Number: 27027027

Customer Name: Akanksha

Account Type: Current

Balance: \$350.0

Create a package `CE` which has two classes `Student` and `intern`. The class `Personal` has members like `USN`, `name`, `sem`. The class `intern` has an array that stores the internal marks stored in five courses of the current semester of the student. Create another package `SE` which has the class `External` which is a derived class of `Student`.

This class has an array that stores SEE marks stored in five courses of the current semester of student. Import the two packages in a file that declares the final set marks of 7 students in all Five courses.

```
→ package CE;
import java.util.*;
public class Student
```

```
{ public int sem;
```

```
String usn;
```

```
String name;
```

```
public void accept()
```

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

~~System.out.println ("Enter sem , USN & name :")~~~~sem = sc.nextInt();~~~~USN = sc.nextLine();~~~~name = sc.nextLine();~~

3

3

public class Internals

{ public int IMF[] = new int [5];
}

package SEE;

import CIE.Student;

public class External extends Student

{ public int SM[] = new int [5] };

import java.util.*;

import SEE.*;

import CIE.*;

public class FinalMarks

{ public static void main (String args[])

int fm[] = new int [5];

Scanner sc = new Scanner (System.in);

System.out.println ("Enter no of students - ");

int n = sc.nextInt();

SEE.External st[] = new SEE.External [n];

CIE.Internals s[] = new CIE.Internals [n];

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

st[i] = new SEE.External();

s[i] = new CIE.Internals();

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

$s(i).accept()$,

for ($\text{int } j=0; j \leq 5; j++$)

{

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

$s(i).im[j] = sc.nextInt()$;

$s(i).sm[5] = sc.nextInt();$

$f[i][j] = s(i).im[j] + s(i).sm[j]$

}

System.out.println("Final marks of " + s(i).name)
for ($\text{int } k=0; k \leq 5; k++$)

{

System.out.println("course " + (k+1) + "=" + f[i][k]);

{

}

~~Output~~

Enter no of students: 1

{

Enter details of 1 student:

enter usn name & sem: IBM22CG027

AKANKSHA 2

Enter internal & external marks: 50 46

Enter internal & external marks: 49 47

Enter internal & external marks: 48 48

Enter internal & external marks: 47 49

Enter internal & external marks: 46 50

Final marks of akanksha

Course 1 = 96

Course 2 = 96

Course 3 = 96

Course 4 = 96

Course 5 = 96

Algorithm

- 1) Start
- 2) Create 2 packages as CSE and SEE
- 3) CSE consists of class which takes the personal details and another class which has an array for internal marks
- 4) Consider a package SEE which has an array to store SEE marks
- 5) Import these packages to the main file
- 6) Create two arrays one for DS of Students present and other array SP which has objects of CSE package
- 7) Access packages with help of these 2 arrays and calculate the final marks ie CSE+SEE marks
- 8) To display the final marks it should be stored in another array and print its content at each.
- 9) STOP

~~25/02/2020~~
02/02/2020

```
C:\Users\bmsce\Desktop\1bm22cs027 ooj\week 6>javac -d . FinalMarks.java

C:\Users\bmsce\Desktop\1bm22cs027 ooj\week 6>java FinalMarks
AKANKSHA SINGA
1BM22CS027
enter no. of students:
2
Enter details of 1
enter usn name and sem:
1BM22CS027
AKANKSHA
2
Enter internal and external marks:
49
49
Enter internal and external marks:
50
50
Enter internal and external marks:
48
48
Enter internal and external marks:
47
47
Enter internal and external marks:
46
46
Final marks of AKANKSHA
Course 1=98
Course 2=100
Course 3=96
Course 4=94
Course 5=92
Enter details of 2
enter usn name and sem:
1BM22CS027
akanksha
1
Enter internal and external marks:
50
46
Enter internal and external marks:
49
47
Enter internal and external marks:
48
48
Enter internal and external marks:
47
49
Enter internal and external marks:
50
46
Final marks of akanksha
Course 1=96
Course 2=96
Course 3=96
Course 4=96
Course 5=96
```

3 Exception Handling

import java.util.Scanner;
class WrongAge extends Exception
{}

 public WrongAge (String message)
 {
 super (message);
 }

}

class Father {

 int fage;

 public Father (int fage) throws WrongAge
 {
 if (fage < 0)
 throw new WrongAge ("Age cannot
 be negative");
 }

}

 this.fage = fage;

}

class Son extends Father

 int sage;

 public Son (int fage, int sage)

 throws WrongAge {

 super (fage);

 if (sage >= fage)

 throw new WrongAge ("son's

3 age must be less than Father's age");
3 this. ~~totally~~ = s. age);
3
3 public class FatherSon
2 {
3 public static void main (String[] args)
4 {
5 Scanner sc = new Scanner (System.in);
6 System.out.print ("Enter father's and
7 son's age");
8 int fa = sc.nextInt();
9 int sa = sc.nextInt();
10 try
11 {
12 Son s = new Son (fa, sa);
13 System.out.println ("Father's age: " +
14 s.getAge());
15 System.out.println ("Son's age: " + s.getAge());
16 }
17 catch (WrongAge e)
18 {
19 System.out.println ("Error: " + e.getMessage());
20 }
21 }

Algorithm

- 1) Start
- 2) Create a class wrongage which is a user exception defined
- 3) Create a class father which takes father's age as parameter & evaluation for greater than zero
- 4) Create a class son which extends father & takes parent parameter of father age & has student age if sage \geq fage it will throw user defined error
- 5) the main function fatherson takes the values of ages from user & prints there age if it is valid
- 6) If the values are invalid it prints an appropriate msg of exception
- 7) Stop.

Output

Enter father's age and son's age : -1 2

Error: age cannot be negative

Enter father's age & son's age : 26 30

Error: son's age must be less than father's age

Enter father's age & son's ag. ? , age

Father's age : 2

Son's age : 1

```
C:\Users\bmsce>cd C:\Users\bmsce\Desktop\1bm22cs027 ooj
```

```
C:\Users\bmsce\Desktop\1bm22cs027 ooj>javac fatherson.java
```

```
C:\Users\bmsce\Desktop\1bm22cs027 ooj>java fatherson
```

```
AKANKSHA SINGA
```

```
1BM22CS027
```

```
Enter father's age and son's age:
```

```
-1
```

```
2
```

```
Error: Age cannot be negative
```

```
C:\Users\bmsce\Desktop\1bm22cs027 ooj>java fatherson
```

```
AKANKSHA SINGA
```

```
1BM22CS027
```

```
Enter father's age and son's age:
```

```
20
```

```
30
```

```
Error: Son's age must be less than Father's age
```

```
C:\Users\bmsce\Desktop\1bm22cs027 ooj>java fatherson
```

```
AKANKSHA SINGA
```

```
1BM22CS027
```

```
Enter father's age and son's age:
```

```
2
```

```
1
```

```
Father's age: 2
```

```
Son's age: 1
```

§ Multi Threading

Class A extends Thread

{
 A public void run()
 {

 while(true)

 System.out.println('BMS college
of Engineering');
 try

 sleep(10000);
 }

 catch (exception e)

 System.out.println('error in class A');

}}}

Class B extends Thread

{
 B public void run()
 {

 while(true)

 System.out.println('CSE');

 try

 sleep(2000);
 }

 catch (exception e)

 System.out.println('error in class B');

}}}

Clean Test

2

Public Static Void Main (String[] args)

2

A a = new A();
B b = new B();
a.start();
b.start();

3

3

Output:

BMS College of Engineering

CSE

CSE

CSE

CSE

GE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

Infinite loop

Algorithms

- 1) Start
- 2) Make 2 classes A & B
- 3) class A prints BMS college of engineering after every 10 seconds
- 4) $\&$ is checked with help of a try catch block
- 5) class B prints GE after every 2 seconds and is checked with try catch block
- 6) The main function creates 2 objects one of class A & other of class B and these two objects point to different classes
- 7) Both the objects in main are called by start method which runs in classes & prints the required output
- 8) Stop

Q2

This
16.02.24

```
C:\Users\bmsce>cd C:\Users\bmsce\Desktop\1bm22cs027 ooj
```

```
C:\Users\bmsce\Desktop\1bm22cs027 ooj>javac multithread.java
```

```
C:\Users\bmsce\Desktop\1bm22cs027 ooj>java multithread
```

AKANKSHA SINGA

1BM22CS027

BMS COLLEGE OF ENGINEERING

CSE

CSE

CSE

CSE

BMS COLLEGE OF ENGINEERING

CSE

CSE

CSE

CSE

CSE

BMS COLLEGE OF ENGINEERING

CSE

CSE

CSE

CSE

CSE

BMS COLLEGE OF ENGINEERING

CSE

CSE

o Write a program to create user interface to perform integer divisions. The user enters two nos in the text fields num1 & num2. The division of num1 & num2 is displayed in the result field when the divide button is clicked. If num1 or num2 were not an integer the program would throw a NumberFormatException. If num2 were zero program would throw an arithmetic exception displaying the exception in a message dialog box.

```
→ import java.awt.*;  
import java.awt.event.*;  
class Swing Demo{  
    Swing Demo(){  
        // Create JFrame container  
        JFrame jfrm = new JFrame ("Divide App");  
        jfrm.setSize (275,150);  
        jfrm.setLayout (new GridLayout());  
        // To terminate on close  
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        JLabel jlab = new JLabel ("Enter dividend & divisor.");  
        JTextField ajtf = new JTextField (8);  
        JTextField bjtf = new JTextField (8);  
        JButton button = new JButton ("Calculate");  
        JLabel err = new JLabel();  
        JLabel ans = new JLabel();
```

```
jlabel_benb = new JLabel();
jlabel_anlab = new JLabel();
jfrm.add(left);
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(amslab);
```

```
ActionListener1 = new ActionListener() {
```

```
    public void actionPerformed(ActionEvent evt) {
```

```
        System.out.println("Action event from  
        a text field");
```

3.

```
ajtf.addActionListener(l);
```

```
bjtf.addActionListener(l);
```

```
button.addActionListener(new ActionListener() {
```

```
    public void actionPerformed(ActionEvent evt) {
```

try {

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

```
    int b = Integer.parseInt(bjtf.getText());
```

```
    int ans = a / b;
```

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

```
    bleb.setText ("\\nB=" + b),  
    ansleb.setText ("\\nAns=" + ans);
```

{

```
catch (NumberFormatException e)
```

{

```
    bleb.setText (" ">,
```

```
    bleb.setText (" ">,
```

```
    ansleb.setText (" ">,
```

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

{

```
catch (ArithmeticException e) {
```

```
    bleb.setText (" ">,
```

```
    bleb.setText (" ">,
```

```
    ansleb.setText (" ">,
```

```
    err.setText ("B Should be non zero");
```

{

{

```
    ifrm.setVisible (true);
```

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

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

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

```
            new SwingDemo ();
```

Output

Enter divider & dividend

34	2	Calculate
----	---	-----------

A=34 B=2 Ans=17

{

34
2
17

