

Program 1:

Java Program to Print an Integer
(Entered by the User)

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

Output: Enter a number: 4
You entered 4 number

Algorithm:-

Step 1 :- Start

Step 2 :- Create a scanner 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

(Start)



/Point Enter a number/



Read the input



Print You entered number!



(Stop)

Program 2:-

Java program to check whether a number is Even or odd.

Algorithm

Step 1: Start

Step 2: Define integer number

Step 3: Print Enter an Integer number

Step 4: Store the input given by user in num

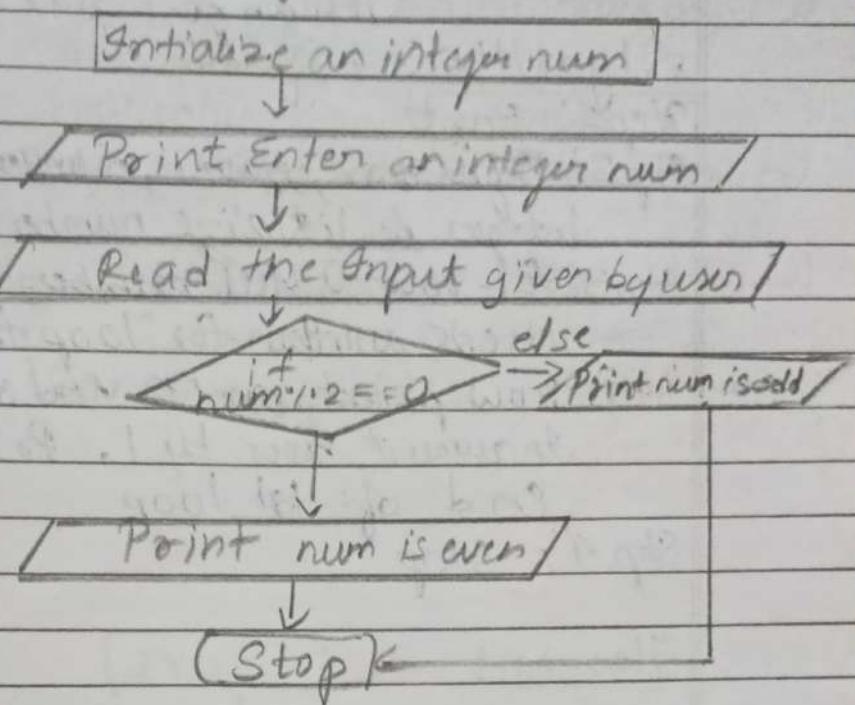
Step 5: Check if num gives remainder 0 when divided by 2

Step 6: If true print number is an even number

Step 7: Else print number is an odd

Step 8: Stop

Flowchart :- (Start)



Input:

```

import java.util.Scanner;
public class JavaExample
  
```

```

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");
}
  
```

Output:

Program 3:- Java program to Print Right Triangle Star Pattern

Algorithm:-

Step 1:- Start

Step 2:- Define row, column, number of rows as integer & initialize number of rows to 8.

Step 3:- for row 0 till number of rows

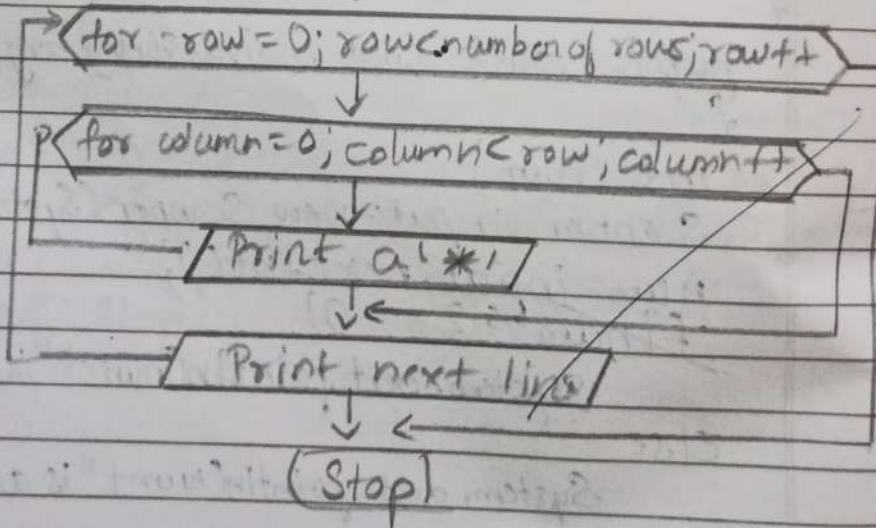
check another for loop from column 0 till row print an *. End of 2nd loop.

Increment row by 1. Print next line
End of 1st loop

Step 4:- Stop

Flowchart (Start)

Define int row, column & number of rows.
Initialize number of rows=8.



Input:-

```

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();
        }
    }
}

```

Output:-

```

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

```

Program4:- Java program to Find Quotient & Remainder

Algorithm:-

Step 1:- Start

Step 2:- Initialize num1 to 15, num2 to 2. Define them integer

Step 3:- Operate quotient = num1 / num2

Step 4:- Operate remainder = num1 % num2

Step 5:- Print Quotient value

Step 6:- Print Remainder value

Step 7:- Stop

(Start)



Define & Initialize num = 15 & num2 = 2

quotient = num1 / num2



remainder = num1 % num2



Print the Quotient value



Print the Remainder value



(Stop)

Input: 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);  
}
```

3

Output: Quotient is : 7

Quotient is : 1

Program 5:- Java Program to multiply two numbers

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

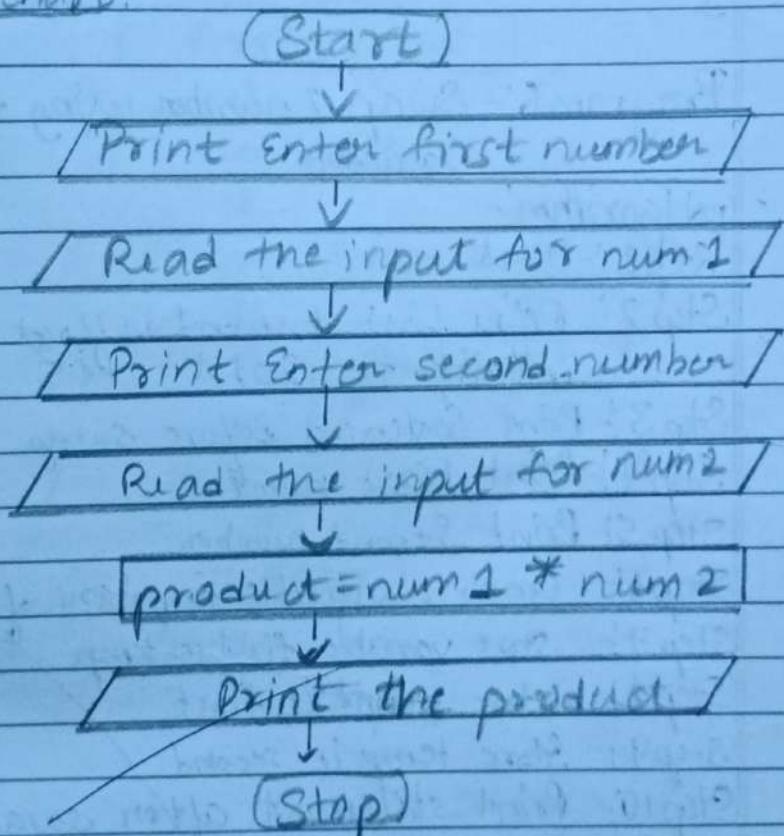
Step 5:- Read input for int num2

Step 6:- calculate product = num1 * num2

Step 7:- print the product value

Step 8:- Stop

Flowchart:-



Output: enter first number

5

enter second number

2

Output: 10

Input: public class Demo

```
public static void main(String args[])
{
    Scanner Scan = new Scanner(System.in);
    System.out.println("Enter first number:");
    int num = Scan.nextInt();
    System.out.println("Enter Second number:");
    int num2 = Scan.nextInt();
    Scan.close();
    int product = num * num2;
    System.out.println("Output: " + product);
}
```

}

Program 6:- Swap 2 numbers using temporary variable

Algorithm:-

Step 1:- Start

Step 2:- Define first & second as float datatype & initialize to 1.20 & 2.45 respectively

Step 3:- Print Statement before swap

Step 4:- Print First number

Step 5:- Print Second number

Step 6:- Create a variable temporary of float datatype

Step 7:- Store variable first in temp

Step 8:- Store second in first

Step 9:- Store temp in second

Step 10:- Print statement after swap

Step 11:- Print First number

Step 12:- Print Second number

Step 13:- Stop

(Start)

Define & initialize first = 1.20f & second = 2.45f

Point 'Before swap'

Print 'First number'

Print 'Second number'

temp = first

first = second

second = temp

Print 'After swap'

Print 'first Number'

Print 'Second number'

Stop

Input:

```
public class Swapnumbers  
{  
    public static void main(String args[]){  
        float first=1.20f, 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);  
    }  
}
```

Output:

- Before swap -

first number = 1.20

Second number = 2.45

- After swap -

first number = 2.45

Second number = 10.1



Problem Quadratic equation (user input)

Algorithm:

Step 1: Start

Step 2: initialise variable a,b,c,d &
~~sqrt a,b,c~~

Step 3: if ($a=0$) print invalid input
 go to Step 6

Step 4: $d = b * b - 4 * a * c$

Step 5: if $d > 0$

print "roots are real"

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

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

print(r_1, r_2) go to step 8

Step 6: if $d < 0$

print ("Roots are imaginary")

There are no real solution

$$r_1 = -b / (2 * a)$$

$$r_2 = \sqrt{|d|} / (2 * a)$$

print ($r_1 + i r_2$)

print ($r_2 - i r_1$) go to step 8

Step 7: If $d = 0$

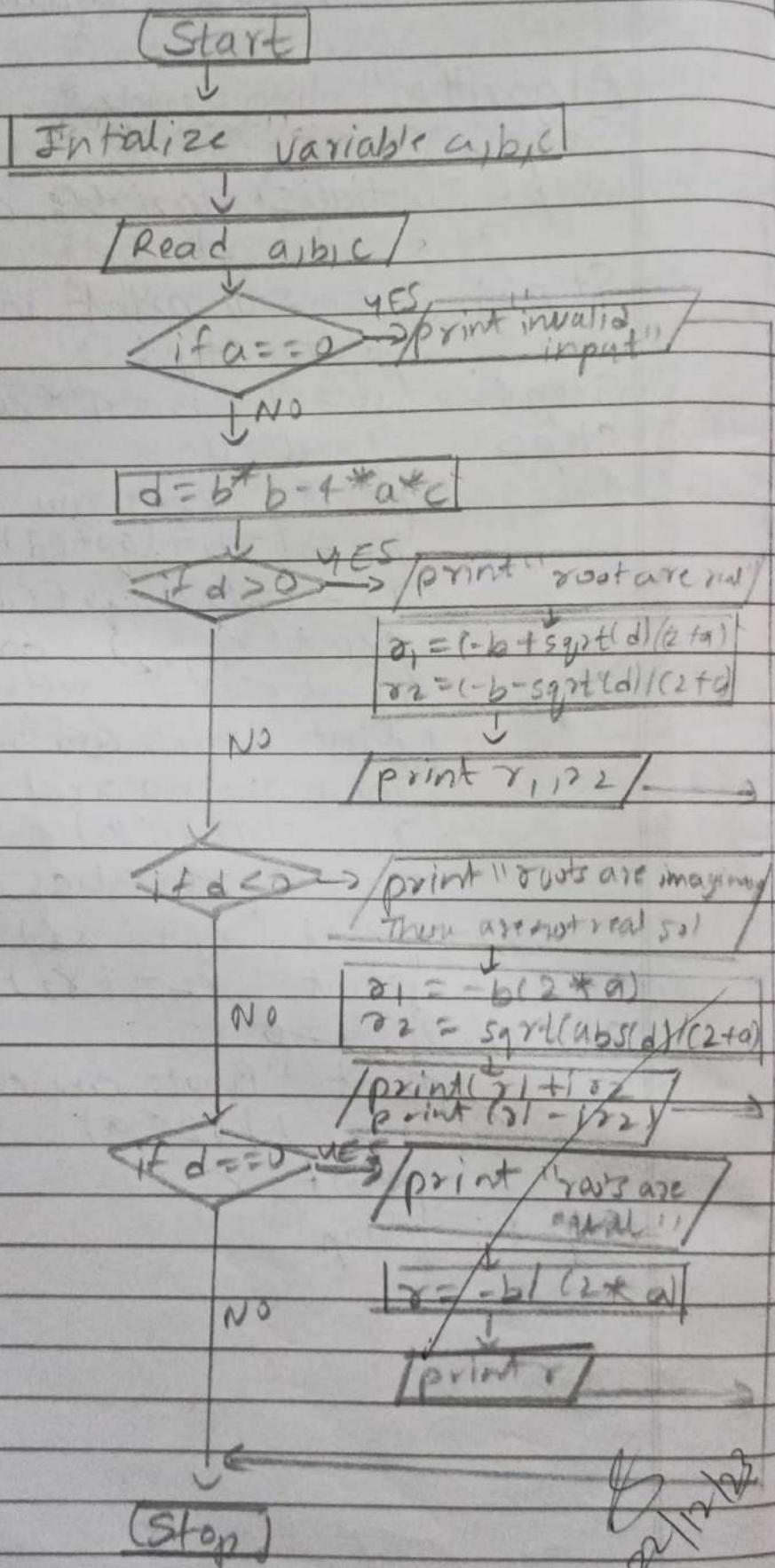
print "Roots are equal"

$$r = -b / (2 * a)$$

print r

Step 8: Stop

Flowchart:



Input:

```
import java.util.Scanner;
class Quadratic
public static void main(String args[]){
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter values a,b & c");
    int a = sc.nextInt();
    int b = sc.nextInt();
    int c = sc.nextInt();
    int d = b*b - 4*a*c;
    double root1, root2, rp, ip;
    if (d == 0)
    {
        System.out.println("Roots are real &
                           equal");
        root1 = root2 = -b / (2*a);
        System.out.println("root1 = " + root1 +
                           "root2 = " + root2);
    }
    else if (d > 0)
    {
        System.out.println("Roots are real
                           and distinct");
        root1 = b + Math.sqrt(d) / (2*a);
        root2 = b - Math.sqrt(d) / (2*a);
        System.out.println("root1 = " + rp +
                           " + ip + " + "root2 = " + rp +
                           " - ip");
    }
}
```

Output:

Enter value a,b,c

2

3

4

roots are imaginary

root1 = 0.0 + 1.198957880281798

root2 = 0.0 - 1.198957880281798

8/2/23

LAB-03

SGPA

```
import java.util.*;
class student {
    String us# by name;
    int n, credit[n], marks[n], sum;
    float sgpa=0;
    Scanner sc = new Scanner(system.in);
    void Accept_Ans_Display() {
        System.out.println("enter your name:");
        name = sc.next();
        System.out.println("enter your usn");
        us# = sc.next();
        for(int i=0; i<n; i++) {
            System.out.println("Enter the credit:");
            credit[i] = sc.nextInt();
            System.out.println("Enter the marks:");
            marks[i] = sc.nextInt();
        }
    }
    void calculate() {
        for(int i=0; i<n; i++) {
            sgpa += credit[i]*marks[i];
            sum += credit[i];
        }
        sgpa = sgpa/sum;
        System.out.println("sgpa is:" + sgpa);
    }
}
```

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

```
    System.out.println ("Enter no of subjects");
    n = sc.nextInt();
    s.Accept_And_Display();
    s.calculate();
}
```

Output: enter your name: Anuj
 enter USN: 46
 enter credit: 4
 enter marks: 80
 enter credit: 3
 enter marks: 100
 SGPA is : 8.857

Algorithm:

Step 1: Start

Step 2: declare variable USN, name, n,
 credit[n], marks[n], sum.

Step 3: Create function accept-and-disp.
 read USN, name, display USN,
 name.

Step 4: Create function calculate

for int i=0 to i<n:

read credit[i], marks[i]

SGPA = credit[i] * (marks[i])

sum = credit[i]

Step 5: SGPA / = sum;

Step 6: display SGPA

Step 7: Stop

(Start)

declare usn, name n, credit[n],
marks[n], sum

read n

Accept - And - Display ()

final usn, name

Student s = new student()

for init:0; i < n; i++)

S. Accept - And - Display ()

S. calculate

read credit[i];

marks[i]

sgpa = sgpa / sum

calculate

display sgpa

< for init:0; i < n; i++ >

End

sgpa = credit[i] * marks[i]

sum += credit[i]

sgpa = sgpa / sum

display sgpa

Lab-03 Program-03 Accessing objects

```
class Boy {  
    String name, address;  
    int age;  
    void tellName();  
    void tellAddress();  
    void tellAge();  
    public static void main(String args)  
}
```

```
Boy obj = new Boy();  
obj.name = "George";  
obj.address = "Los Angeles, USA";  
obj.age = 23;  
System.out.println("Name of the boy is " + obj.name);  
System.out.println("Address of boy is " + obj.address);  
System.out.println("Age of boy is " + obj.age);
```

Output:

Name of the boy is: George
Address of boy is: Los Angeles, USA
Age of boy is: 23

✓ 12/11/12

Lab-04

Program 03:

Create a class Book which contains four members: name, author, price, numPages. Include a constructor to set the values for the members. Include method to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

Program

```
import java.util.Scanner;  
class Book{  
    private String name;  
    private String author;  
    private double price;  
    private int numPages;  
    public Book(String name, String author, double price, int numPages)  
    {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.numPages = numPages;  
    }  
    public void setName(String name){  
        this.name = name;  
    }  
    public String getName(){  
        return name;  
    }  
    public void setAuthor(String author){  
        this.author = author;  
    }  
}
```

```
public String getAuthor() {  
    return author;
```

{

```
public void setPrice(double price) {  
    this.price = price;
```

{

```
public double getPrice() {  
    return price;
```

{

```
public void setNumPages(int numPages) {  
    this.numPages = numPages;
```

{

```
public int getNumPages() {  
    return numPages;
```

{

```
public String toString() {
```

```
    return "Book Details:\nName:" + name +  
        "\nAuthor:" + author + "\nPrice:" + price +  
        "\nnumPages" + numPages;
```

{

{

```
public class BookTest {
```

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

~~Scanner sc = new Scanner(System.in);~~~~System.out.println("Enter the number of books");~~~~int n = scanner.nextInt();~~~~Book[] books = new Book[n];~~~~for (int i=0; i<n; i++) {~~ ~~System.out.println("Enter details for book
 +(i+1)+":");~~ ~~scanner.nextLine();~~ ~~System.out.println("Name:");~~

```
String author = sc.nextLine();
System.out.print("Book: $");
double price = sc.nextDouble();
System.out.print("Number of pages: ");
int numPages = sc.nextInt();
books[i] = new Book(name, author, price,
numPages);

for(int i=0; i<n; i++){
System.out.println("In Details for Book" +
(i+1) + ". . .");
System.out.println(books[i]);
}

scanner.close();
```

Output: Enter the number of books: 1

Enter details for Book 1:

Name: The Catcher in the Rye

Author: J. D. Salinger

Price: ₹ 300

Number of Pages: 224

~~Details for Book 1.~~

~~Book Details~~

Name: The Catcher in the Rye

Author: J. D. Salinger

Price: ₹ 300

Number of Pages: 224

Program 4:

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

Program:

```

abstract class Shape{
    protected int side1;
    protected int side2;
    public Shape(int side1,int side2){
        this.side1=side1;
        this.side2=side2;
    }
    public abstract void printArea();
}

class Rectangle extends Shape{
    public Rectangle(int length,int width){
        super(length,width);
    }
    public void printArea(){
        int area=side1*side2;
        System.out.println("Area of Rectangle: "+area);
    }
}

```

class Triangle extends Shape

```
public Triangle (int base, int height) {
    super(base, height);
```

3:

```
public void printArea() {
```

```
double area = 0.5 * side1 * side2;
```

```
System.out.println("Area of triangle"
    + area);
```

3:

class Circle extends Shape

```
public Circle (int radius) {
    super(radius, 0);
```

3:

```
public void printArea() {
```

```
double area = Math.PI * side1 * side2;
```

```
System.out.println("Area of circle"
    + area);
```

3:

public class ShapeTest

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

```
Rectangle rectangle = new Rectangle(4, 5);
```

```
Triangle triangle = new Triangle(3, 8);
```

```
Circle circle = new Circle(6);
```

```
rectangle.printArea();
```

```
triangle.printArea();
```

```
circle.printArea();
```

3:

3:

Output: Area of Rectangle: 20

Area of Triangle: 12.0.

Area of Circle: 113.0973552923255

Algorithm:

Step 1: Start

Step 2: Declare private variables name, price, author, numof Pages with suitable datatypes.

Step 3 : create constructor to initialize these variables

Step 4: create get & set for each variable

Step 5: create toString function with parameters

Step 6 : Read all variables

Step 7: call toString and print values

Step 8: Stop.

Start

void details()

private String name, author
private int num Pages
private float price

enter details of book i+1

read name, author,
num pag /price

read n

Book[] books = new Book[n]

for i=0; i<n; i++

void Details()

toString()

toString()

Details for Book i+1

display books[i].name

display books[i].author

display books[i].page.num

display books[i].price

Stop

AlgorithmStep 1: StartStep 2: declare side 1, side 2Step 3: initialize side 1, side 2 using constructor & classStep 4: class rectangle, triangle, circle extends to print areaStep 5: print rectangle area, triangle area, circle areaStep 6: exit

Shape(side1, side2)

Start

this.side1 = side1

this.side2 = side2

side1, side2

rectangle extends shape

rectangle r = new rectangle(4, 5)
triangle t = new triangle(3, 8)
circle c = new circle(6)

rectangle(int length, int width)

super(length, width)

rectangle.printArea()

triangle.printArea()

circle.printArea()

triangle extends shape

print area(1)

triangle(base, height)

print length * width

super(base, height)

print $\frac{1}{2} * \text{radius}^2$

print 0.5 * base * height

circle extends shape

exit

circle(radius)

super(2, radius, 0)

Friday
19-01-2024

SURYA Gold

Date _____

Page _____

Lab - 05

Program to create a class bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account name & 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:

- a) Accept deposit from customer and update the balance
- b) Display the balance
- c) Compute and deposit interest
- d) Permit withdrawal and update the balance. Check for the minimum balance, impose penalty if necessary and update their balance.

Q) Execute the following program

```
public class Cart {
```

```
    private String itemNam;
```

```
    private int price;
```

```
    private int quantity;
```

```
    public void setItemName(String ItemName){
```

```
        this.itemName = itemName;
```

```
}
```

```
    public String getItemName(){
```

```
        return itemName;
```

```
}
```

```
    public void setPrice(int price){
```

```
        this.price = price;
```

```
}
```

```
    public int getPrice(){
```

```
        return price;
```

```
}
```

~~public static void main(String[] args){~~ ~~Cart obj = new Cart();~~ ~~obj.setItemName("Butter");~~ ~~obj.setPrice(50);~~ ~~System.out.println("The details we
have set are: ");~~~~System.out.println(obj.getItemName());~~~~System.out.println(obj.getPrice());~~~~}~~

Output:

Button
50

Algorithm:

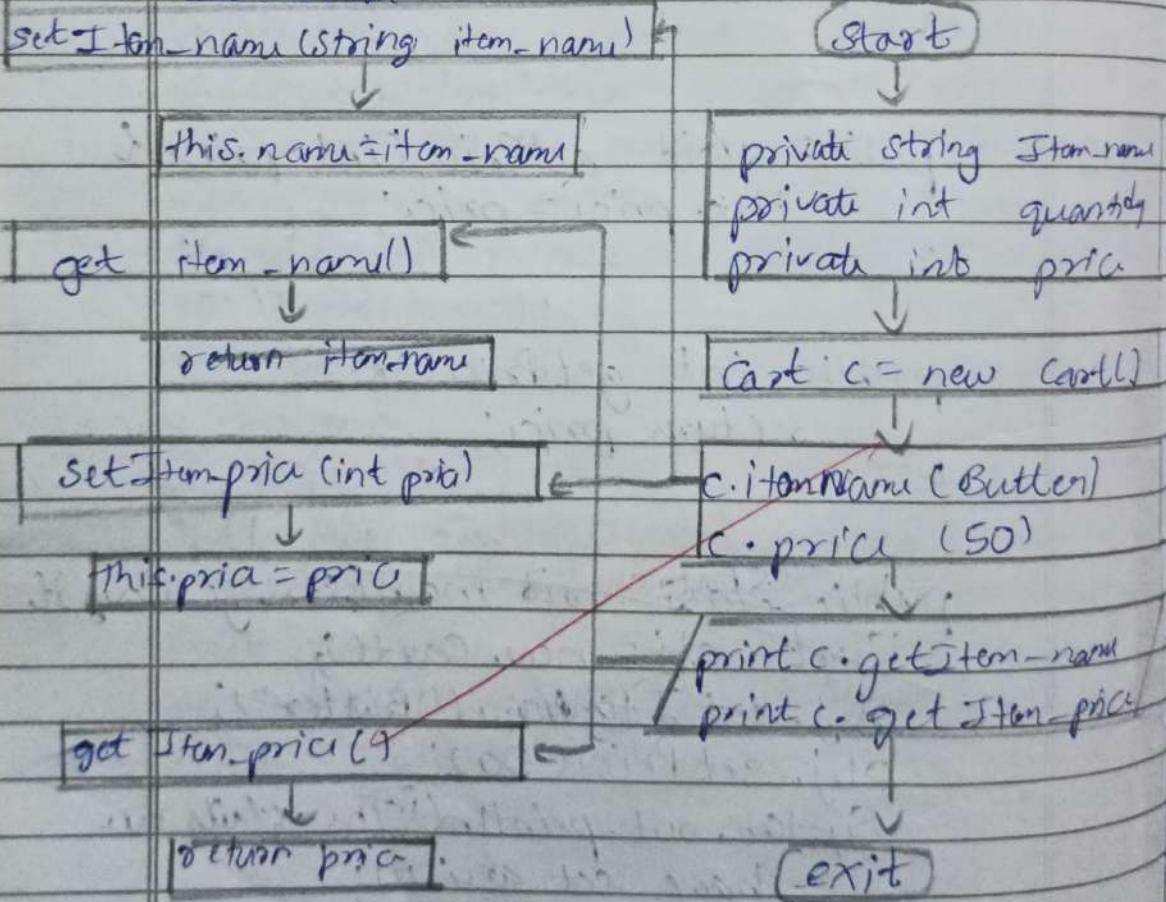
Step 1: Start

Step 2: declare quantity, item-name, price (private)

Step 3: create getters & setters for
item-name, price.

Step 4: print item-name

Step 5: exit

Flowchart:

```
import java.util.Scanner;  
class Account
```

```
    String customerName;  
    int accountNumber;  
    String accountType;  
    double balance;
```

```
Account(String name, int acctno, String  
        accttype, double bal)
```

{

```
    name = customerName;  
    acctno = accountNumber;  
    accttype = accountType;  
    bal = balance;
```

}

```
void displayBalance()
```

{

```
printf("The balance for the account  
accountnumber %d : %.2f", bal)
```

```
class curAcc extends Account
```

```
    double minBalance;
```

```
    double serviceCharge;
```

```
curAcc(String name, int accNumber,
```

```
        double initialBalance, double minBalance,
```

```
        double serviceCharge)
```

{

```
super(name, accNumber, "current");  
initialBalance);
```

```
this.minBalance = minBalance;
```

```
this.serviceCharge = serviceCharge;
```

}

```
void withdraw(double amount)
```

{

```
    if (balance - amount < minBalance) {
        System.out.println("Withdraw not permitted. Minimum balance not maintained.");
        imposeServiceCharge();
    }
```

else {

```
    balance -= amount;
```

```
    System.out.println("$" + amount +
        " withdrawn successfully!");
```

```
    displayBalance();
}
```

}

```
private void imposeServiceCharge()
```

```
    System.out.println("Service charge of $" + serviceCharge +
        " imposed.");
```

```
    balance -= serviceCharge;
```

```
    displayBalance();
}
```

}

```
class SavAcct extends Account {
```

```
    double interestRate;
```

~~SavAcct(String name, int accNumber, double initialBalance, double interestRate) {~~

~~super(name, accNumber, "Savings", initialBalance,~~

~~this.interestRate = interestRate;~~

}

```
void depositInterest() {
```

```
    double interest = balance * interestRate / 100;
```

```
    balance += interest;
```

```
    System.out.println("$" + interest + "
```

```
    withdrawn successfully");
```

displayBalance();

getac

void withdraw(double amount){

if (balance >= amount){

balance -= amount;

System.out.println("₹ " + amount +
" withdrawn successfully");

displayBalance();

} else

System.out.println("Withdrawal
not permitted Insufficient Balance");

}

3

public class Bank{

public static void main(String[] args){
Scanner sc = new Scanner(System.in);
Current currentAccount = new
CurAcc ("John Smith", 67890,
1000.0, 5000, 10);

SavAcc savingAccount = new

SavAcc ("John Doe", 12345,
1000.0, 5000, 50);

currentAccount.displayBalance();

currentAccount.withdraw(4000);

savingsAccount.displayBalance();

savingsAccount.depositInterest();

savingsAccount.withdraw(1000.0);

3

3

Output.

Balance for account 12345: Rs 1000.0
Rs. 200.0 withdrawn successfully
Balance for account 12345: Rs 800.0

Balance for account 67890: Rs 2000.0
Interest of Rs 100.0 deposited.
Balance for account 67890: Rs 2100.0
Rs 1500.0 withdrawn successfully
Balance for account 67890: Rs 600.0

CGO
19/11/14

Algorithm:

Step 1:

02-02-2024

Lab-06

Create a package CIE 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 the current semester of the student. Create another package SEE which has the class External which is a derived class of student. This class has an array that stores the SEE marks scored in 5 courses of the current semester of the student. Import the 2 packages in a file that declares the final marks of n students in all subjects.

Sample program:

Create a package Forest & create tiger class in it

package forest;

public class Tiger

public void getdetails(String name, int age)

System.out.println("name of the
Tiger is: " + name);

System.out.println("age of the Tiger is:
" + age);

}

}

import. forest. Tiger;
class main

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

```
Tiger T1 = new Tiger();
T1.getDetails ("Leo", 5);
```

{}

O/P → name of the Tiger is : Leo
age of the Tiger is : 5

Main program :

1st package CIE;
public class student {
 String USN;
 String name;
 int sem;

{}

imported
public class Internals extends student
 public int[] internal = new int[5];

{}

2nd package SEE;
import CIE.student;
public class Externals extends CIE.student
 public int[] external = new int[5];

{}

```
import CIE.student;
import SEE.external;
import java.util.Scanner;

public class Filenamet
    public static void main(String args[])
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter number
            of students");
        int n = s.nextInt();
        int finalmarks[][] = new int
            [n][5];
        for (int i = 0; i < n; i++)
        {
            System.out.print("Student" +
                (i + 1) + ":");

            System.out.print("Enter internal
                marks for 5 courses");
            Internals internals = new Internals();
            for (int j = 0; j < 5; j++)
            {
                internals.internalMarks[i][j] =
                    scanner.nextInt();
            }

            System.out.print("Enter external
                marks for 5 courses");
            Externals external = new Externals();
            for (int j = 0; j < 5; j++)
            {
                external.externalMarks[i][j] =
                    scanner.nextInt();
            }
        }
    }
}
```

for (int j = 0; j < 5; j++) {
 Final Marks [i][j] =
 intervals. Enter Marks [j].
 external, external Marks [j];

3

3

```
System.out.println("Final Marks.");  

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

    System.out.println("Student " + (i + 1) + "
```

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

```
System.out.println(FinalMarks[i][j] + " ");
```

3

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

3

```
scanner.close();
```

3

~~Q3~~

O/P: Students

Enter the number of students: 1

Enter the name: Aashirwad
 Enter the ID: 2023030252 Enter internal marks 5 courses:

10

20

30

40

50

Enter external marks 5 courses:

10

20

30

40

50

Student 1: 20 40 60 80 100

~~Q3 part 2~~

Algorithm :-

Step 1: start

Step 2: create package CIE enter
username & sun

Step 3: input SEE, CIE
util in main

Step 4: import package SEE
and create array interval

Step 5: create method

calculate final marks

inputs \rightarrow marks \times twice

$$\text{final marks} = \frac{\text{input} + \text{input}}{2}$$

Step 6: for($\text{int } i = 0; i < n; i++$)
 $\quad \quad \quad \text{print}($ final marks $)$

Step 7: stop

Friday
16-02-24

SURYA Gold

Date:

Page:

Program - 07

Write a program that demonstrates handling of exception in inheritance term. Create a base class called "Father" and derived class called "son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAgeException when the input age < 0. In Son class, implements a constructor that takes both father & son's age & throws an exception if son's age is \geq father's age.

Input:-

```
class WrongAgeException extends Exception  
public WrongAgeException(String message){  
    super(message);  
}
```

}

```
class Father {
```

```
    int d-age;
```

```
    public Father(int a) throws WrongException  
    if(a < 0){
```

```
        throw new WrongAgeException("Age is less than  
        zero");
```

```
    }  
}
```

```
class son extends Father {
```

```
    int s-age;
```

```
    public Son(int d-age, int s) throws WrongAgeException  
    {
```

```
        super(d-age);  
    }
```

if ($d_age < s$)

{

throw new WrongAgeException(" Father
age cannot be less than son.");

{

s.age = s;

{

{

public class Main

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

System.out.println("Hello World");

try {

Son s1 = new Son(-1, 20);

{

catch(WrongAgeException e){}

System.out.println("Exception: " +

e.getMessage());

{

{

{

Output:

Hello World!

~~Age~~

~~Exception: Age is less than zero!~~

Program - 08

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

class ps1 implements Runnable

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

```
for(int i=0; i<5; i++)
{
```

```
System.out.println("BMS COLLEGE OF ENGINEERING");
    
```

```
Thread.sleep(10000);
}
```

```
catch(Exception e)
{
```

```
e.printStackTrace();
}
```

```
}
```

```
}
```

class ps2 implements Runnable

```
{
```

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

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

```
Thread.sleep(2000);
}
```

```
catch(Exception e)
```

```
}
```

3 e.printStackTrace();

3

3

public class

{ public static void main(String args[]){
 PS1 s1 = new PS1();
 PS2 s2 = new PS2();
 Thread t1 = new Thread(s1);
 Thread t2 = new Thread(s2);
 t1.start();
 t2.start();

2

Outputs:

CSE

BMS COLLEGE OF ENGLISH

CSE

CSE

CSE

CSE

BMS COLLEGE OF ENGLISH

BMS COLLEGE OF ENGLISH

BMS COLLEGE OF ENGLISH

BMS COLLEGE OF ENGLISH

Program 7: Algorithm

Step 1: Start

Step 2: Create class WrongAge which extends Exception to display message when exception is called

Step 3: Create Father class

Step 4: Initialize a variable FatherAge
Step 5: Create constructor which throws exception & message if age < 0 & stores the age

Step 6: Create class Son which extends Father class.

Step 7: Initialize variable sonAge.

Step 8: Create constructor which throws exception & message if sonAge \geq FatherAge & stores sonAge

Step 9: Create class FatherSon with main method

Step 10: read fatherAge & sonAge after initializing variables fatherAge & sonAge

Step 11: in try block call Son class constructor & print sonAge & fatherAge

Step 12: On exception print message

Step 13: Stop

Program 8 - Algorithm

Step 1: start

Step 2: create class ps1 extends Thread

Step 3: create constructor which initializes 2 variables t1

Step 4: print BMS college of Engineering with while loop.

Step 5: In try box sleep(10000) means 10 sec sleep in same while loop

Step 6: create class ps2 extends Thread

Step 7: create constructor which initializes 2 variables t2

Step 8: Print CSE with while loop

Step 9: In try box sleep(2000) means 2 sec sleep in same while loop

Step 10: start the threads

Step 11: stop

~~88
8
16 / 2 / 2²~~

Friday
23/02/24

Program - 09

Write a program that creates a user interface divisions. The user enters 2 numbers in the text fields, Num1 & Num2. The division of Num2 & Num1 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, the program would throw an ArithmeticException. Display the exception in a message dialog box.

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

```
class SwingDemo1 {
```

```
    SwingDemo1() {
```

```
        JFrame jfrm = new JFrame("Divide");  
        jfrm.setSize(275, 150);  
        jfrm.setLayout(new FlowLayout());  
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

~~```
JLabel jlab = new JLabel("Enter
the divisor and dividend")
```~~~~```
JTextField ajtf = new JTextField();  
JTextField btf = new JTextField();
```~~~~```
JButton button = new JButton("Calculate")
```~~~~```
JLabel err = new JLabel();
```~~~~```
JLabel eor = new JLabel();
```~~~~```
JLabel err2 = new JLabel();
```~~~~```
JLabel anslab = new JLabel();
```~~

```

jfrm.add(lerd);
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

```

ActionListener l = new ActionListener() {  
 public void actionPerformed(ActionEvent evt) {

try {

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

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

int ans = a / b;

alab.setText("\nA = " + a);

blab.setText("\nB = " + b);

anslab.setText("\nAns = " + ans);

}

Aithmetic

catch (ArithmeticException e) {

alab.setText(" " );

blab.setText(" " );

anslab.setText(" " );

err.setText("B should be Non zero");

}

} ;

jfrm.setVisible(true);

}

catch (NumberFormatException e) {

alab.setText(" " );

blab.setText(" " );

anslab.setText(" " );

err.setText("Enter only Integers");

```

public static void main(String args[])
{
 SwingUtilities.invokeLater(new Runnable()
 {
 public void run()
 {
 new SwingDemo1();
 }
 });
}

```

## 1. imports

- java.awt.\*; Provides classes for creating graphical user interfaces (GUIs) in Java.
- java.awt.event.\*; Contains classes for handling events like button clicks & texts filling.

## 2. Class Swing Demo

construct. swingDemo1();

Creates a JFrame window titled "Divider App" with dimensions 275x150 pixels.

sets the layout to FlowLayout, arranging components in left-to-right flow.

Specifies that the program should terminate when the window is closed.

Creates labels, text fields, a button, and error labels for user input & output. Adds these components to the frame in a specific order.

Attaches actions listeners to the text fields & button to handle user interactions.

## Main method:

Uses SwingUtilities.invokeLater() to ensure all creation happens on the event dispatching thread for thread safety. Creates an instance of the swingDemo class in the GUI setup.

### 3. Action Listener

ActionListener for text fields: Prints a message to the console when a text field is interacted with. ActionListener for button: Attempts to parse the text from the text fields as integers (a and b).

If successful, performs division (a/b) and displays the results in corresponding labels. Catches potential exceptions.

NumberFormatException: Displays an error message if non-integer values are entered.

ArithmaticException: Displays an error message if division by zero is attempted.

### 4. Visualizing the Application:

When you run the code, a simple GUI window appears with:

A label prompting for input.

A "Calculate" button

Labels to display the entered values & the result. An error label for displaying error messages.

### Outputs:

|                                                                            |                         |
|----------------------------------------------------------------------------|-------------------------|
| <input type="checkbox"/> Divider app - <input checked="" type="checkbox"/> |                         |
| B should be Non-zero!                                                      |                         |
| Enter the dividend and divisor                                             |                         |
| <input type="text"/> 50                                                    | <input type="text"/> 10 |
| calculate $\frac{A}{B}$                                                    |                         |

8  
3212a