

1) Print an Integer (entered by user)

```
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);
    }
}
```

Output:

Enter a number:

2

You entered : 2

2) Check whether a no. is even or odd

```
import java.util.Scanner;
public class JavaExample {
    public static void main (String[] args) {
        int num;
        System.out.print ("Enter integer: ");
        Scanner input = new Scanner (System.in);
        num = input.nextInt ();
        if (num % 2 == 0)
            System.out.println (num + " is even");
        else
            System.out.println (num + " is odd");
    }
}
```

O/p:

Enter integer:

6

6 is even

3) Paint right hand star pattern:

```
public class Star {
```

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

```
        int r, c, n = 8;
```

```
        for (r = 0; r < n; r++) {
```

```
            for (c = 0; c <= r; c++) {
```

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

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

```
}
```

```
}
```

O/p:

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

4) print find Quotient & Remainder:

```
public class example {
    public static void main (String [] args) {
        int n1 = 15, n2 = 2;
        int q = n1 / n2;
        int r = n1 % n2;
        System.out.println ("Quotient is : +q");
        System.out.println ("Remainder is : +r");
    }
}
```

D/p: Enter two numbers:
 Quotient is : 7
 Remainder is : 1

5) Multiply 2 nos.

```
import java.util.Scanner;
public class multiply {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print ("Enter 1st no.: ");
        int n1 = sc.nextInt();
        System.out.print ("Enter 2nd no.: ");
        int n2 = sc.nextInt();
        sc.close();
        int p = n1 * n2;
        System.out.println ("Output: " + p);
    }
}
```

D/p: Enter 1st no.: 2
 Enter 2nd no.: 10
 Output: 20

6) Swap using temporary variable

```
public class swap {  
    public static void main (String [] args) {  
        float first = 1.20f, second = 2.45f;  
        System.out.println ("-- Before swap --");  
        System.out.println ("First no. = " + first);  
        System.out.println ("Second no. = " + second);  
        float temporary = first;  
        first = second;  
        second = temporary;  
        System.out.println ("-- After swap --");  
        System.out.println ("First no. = " + first);  
        System.out.println ("Second no. = " + second);  
    }  
}
```

o/p: --Before Swap--

First no. = 1.20 Second no = 2.45 --After Swap-- First no. = 2.45 Second no.
= 1.20

① Algorithm:

Step 1: Start

Step 2: Create a reader instance to take input

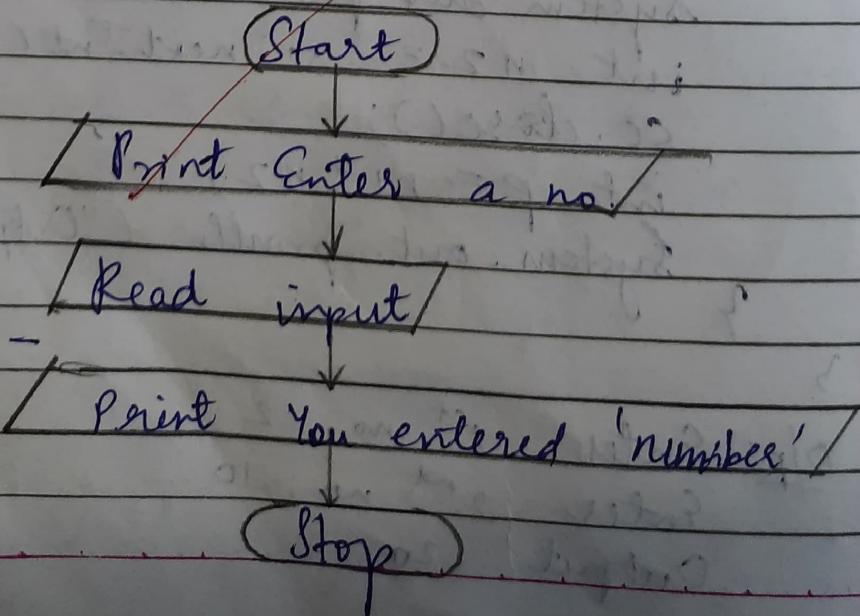
Step 3: Print Enter no.

Step 4: Read no. obtained by user

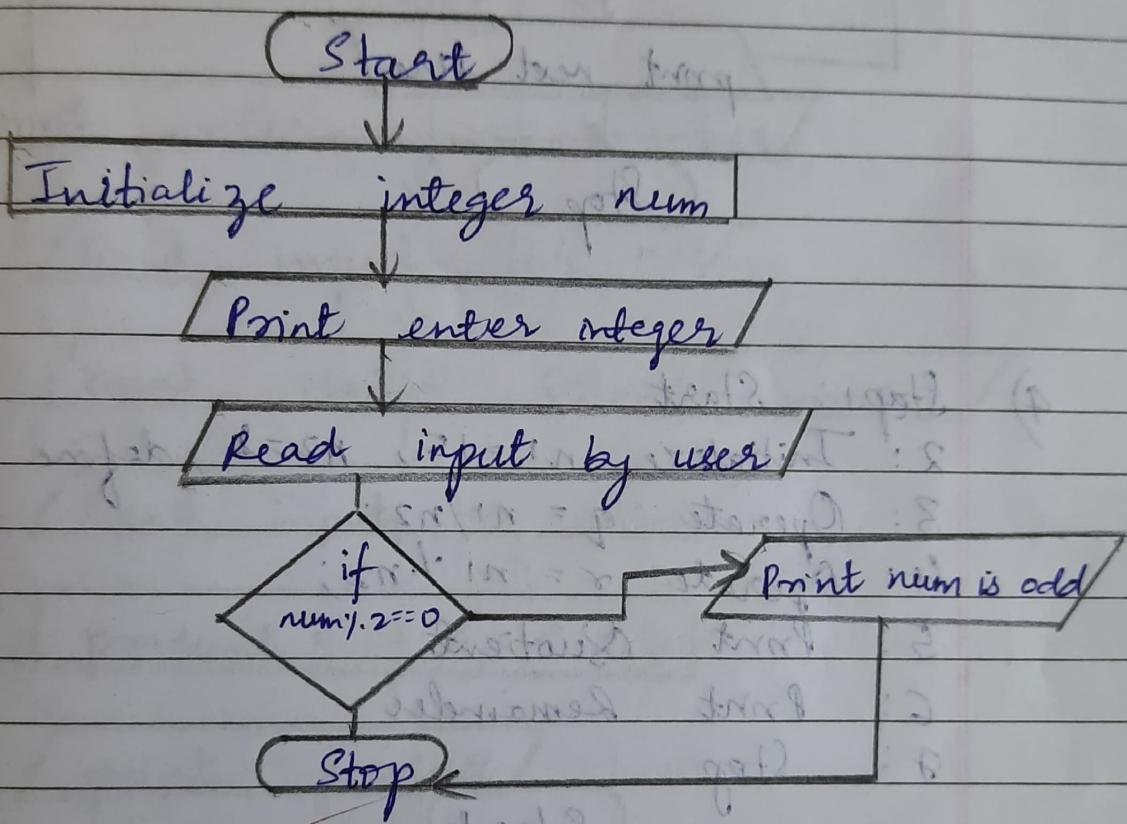
Step 5: Print you entered (number)

Step 6: Stop

Flowchart:



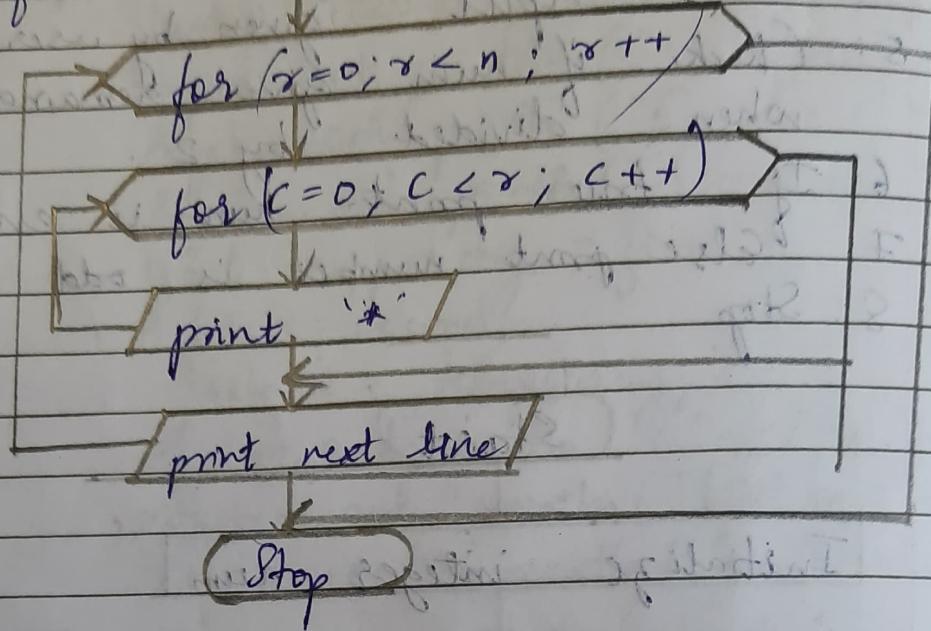
- ② Step 1: Start
- 2: Define integer num
 - 3: Print Enter integer no.
 - 4: Store input given by user in num
 - 5: Check if num gives remainder 0 when divided by 2.
 - 6: If true print num is even
 - 7: Else print number is odd
 - 8: Stop



- ③ Step 1: Start
- 2: Define r, c, n as integers & initialize $n=8$.
 - 3: Run a loop for r 0 to n
check whether another loop from c 0 to r
Print a *.
 - Increment row by 1. Loop continues.
 - 4: Stop

Start

Define ~~s, c, n~~ & initialize $n = 8$



4) Step 1: Start

2: Initialize $n_1 = 15$, $n_2 = 2$ define as integer

3: Operate $q = n_1 / n_2$;

4: Operate $r = n_1 \% n_2$;

5: Print Quotient

6: Print Remainder

7: Stop

Start

Define & initialize $n_1 = 15$ & $n_2 = 2$

$q = n_1 / n_2$

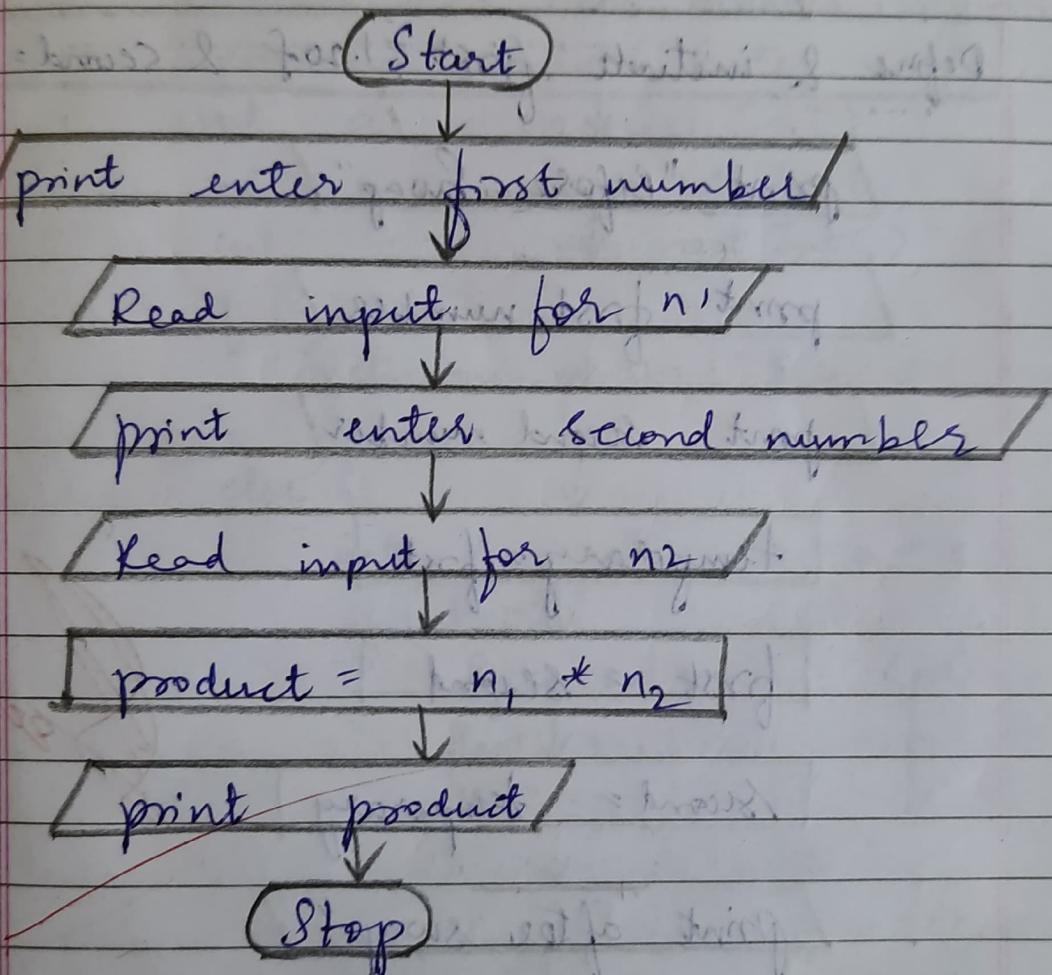
$r = n_1 \% n_2$

Print Quotient

Print Remainder

Stop

- 5) Step1 : Start
- 2: print enter first number
- 3: Read input 'integer n1' from user
- 4: Print enter second number
- 5: Read input for int n2
- 6: Calculate product = $n_1 * n_2$
- 7: print product
- 8: Stop



- 6) Step1 : Start
- 2: Define first & second as float datatype & initialize to 1.20f & 2.45f
- 3: Print "Before Swap"
- 4: Print 1st no.
- 5: Print 2nd no.
- 6: Create a temporary variable of float
- 7: Store first in temporary

- 8: Store second in first
- 9: Store temporary in second
- 10: Print "After Swap"
- 11: Print 1st no.
- 12: Print 2nd no.
- 13: Stop

(Start)

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

print "Before Swap"

print "first number"

print "second number"

temporary = first

first = second

second = temporary

print "after swap"

print "first number"

print "second number"

22/12/23

Page No. 9
Date: 22/12/23

Quadratic Equation Program

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

public class quad {
    public static void main (String[] args) {
        Scanner in = new Scanner (System.in);
        System.out.println ("Enter coefficients:");
        int a = in.nextInt();
        int b = in.nextInt();
        int c = in.nextInt();
        if (a==0) {
            System.out.println ("Invalid input");
        }
        else {
            int d = b*b - 4*a*c;
            if (d>0) {
                System.out.println ("Roots are real");
                float r1 = (float)(-b + sqrt(d)) / (2*a);
                float r2 = (float)(-b - sqrt(d)) / (2*a);
                System.out.println (r1);
                System.out.println (r2);
            }
            elseif (d<0) {
                System.out.println ("Roots are imaginary");
                System.out.println ("There are no real solutions");
                float r1 = (float) -b / (2*a);
                float r2 = (float) sqrt (abs(d)) / (2*a);
                System.out.println (r1 + " + i " + r2);
                System.out.println (r1 + " - i " + r2);
            }
        }
    }
}
```

else {

System.out.println("Roots are equal");
float r = (float) -b / (2 * a);
System.out.println(r);

}

}

O/p (i) Enter coefficients: 0 0 1 1

Invalid input.

(ii) Enter coefficients: 1 2 1

Roots are equal: -1

(iii) Enter coefficients: 1 -6 5

Roots are real: 5 1

(iv) Enter coefficients: 5 1 5

Roots are imaginary. There are no real solutions.

$-0.1 + i0.99498737$ $-0.1 - i0.99498737$

Algorithm:

Step 1: Start

Step 2: initialise variable a, b, c, d & read
a, b, c

Step 3: if ($a=0$) print "invalid input" goto 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) goto Step 8.

Step 6: if $d < 0$

print ("Roots are imaginary
There are no real solutions")

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

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

print ($r_1 + i r_2$)

print ($r_1 - i r_2$) goto step 8.

Step 7: if $d=0$

print " Roots are equal"

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

print r

Step 8: Stop.

Flowchart:

Start

Initialize variables a,b,c

Read a,b,c

if ($a=0$)

Yes

print "invalid input"

No

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

if ($d > 0$)

Yes

print "roots are real"
 $r_1 = (-b + \sqrt{d}) / (2 * a)$
 $r_2 = (-b - \sqrt{d}) / (2 * a)$

No

print r_1, r_2

if $d < 0$

Yes

print "Roots are imaginary
There are no real solutions"

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

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

print $(r_1 + i r_2)$
 print $(r_1 - i r_2)$

if $d=0$

Yes

print "roots are equal"

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

print r

End

29/12/23

Date: _____

SGPA Calculation.

```
import java.util.Scanner;  
  
public class Student {  
    String usn;  
    String name;  
    private static int credits[] = {4, 4, 4, 3, 3, 2, 1};  
    int noOfSub = 8;  
    private int marks[] = new int[noOfSub];  
    Scanner sc = new Scanner(System.in);  
  
    Student() {  
        System.out.println("Enter your details:");  
        details();  
        getMarks();  
        display();  
    }  
}
```

```
public void details() {  
    System.out.println("Enter your USN:");  
    usn = sc.next();  
    System.out.println("Enter your name:");  
    name = sc.next();  
}
```

```
public void getMarks() {  
    System.out.println("Enter marks according to credits:");  
    for (int i = 0; i < noOfSub; i++) {  
        marks[i] = sc.nextInt();  
    }  
}
```

```
double sgpa() {  
    double sgpa = 0, temp = 0;  
    for (int i=0; i<no of Sub; i++) {  
        if (marks[i] >= 40) {  
            if (marks[i] == 100) {  
                temp += credits[i] * ((int)(marks[i]/10));  
            }  
            else if (marks[i] < 100) {  
                temp += credits[i] * ((int)(marks[i]/10)) + 1;  
            }  
            else if (marks[i] < 40) {  
                temp += 0;  
            }  
        }  
        else {  
            temp += 0;  
        }  
    }  
    sgpa = temp / 22;  
    return sgpa;  
}
```

```
void display() {  
    System.out.println("USN :" + usn);  
    System.out.println("Name :" + name);  
    System.out.println("SGPA is :" + sgpa());  
}
```

```
public static void main (String[] args) {  
    Student s1 = new Student();  
}
```

O/p: Enter your details :

Enter marks according to credits:

99 94 92 87 83 96 83 90

USN : 1BM22CS010

Name : Abubakar

SGPA is : 9.6818181818182

Algorithm: Step 1 Start

Step 2: Create class Student and initialize usn and name as string.

Step 3: Initialise credits array as {4, 4, 4, 3, 3, 2, 1, 1}

Initialise noOfSub = 8

Initialise marks array with size noOfSub

Step 4: Create constructor Student

Print (Enter your details:)

Call functions details(), get_marks(), display,

Step 4: Create method details()

print (Enter usn) print (Enter name)

Take user input for USN and name

Step 5: Create method get_details()

print (Enter your marks)

Take user input for marks.

Step 6: Create method double sgpa()

Initialize sgpa=0, temp=0 as double.

Run loop for (i=0, i < noOfSub, i++)

if marks >= 40,

if marks = 100, $\text{sgpa} = \frac{\text{temp} + \text{credits} * (\text{marks})}{10}$

else if marks < 100, $\text{sgpa} = \frac{\text{temp} + \text{credits} * (\text{marks})}{10}$

else temp += 0

else temp = temp + 0. sgpa = $\frac{\text{temp}}{22}$.

Step 7: Create method display()

print USN, name and call sgpa().

Step 8: Create main function

and make object s1 of Student.

Step 9: Stop

Ais
29/12/23

12/1/24

66
Page No. 18
Date: _____

- 3) Create class book which consists of name, author, price, num-pages. Include constructor, set values. Include set, get details, toString() method.

Soln

```
import java.util.Scanner;
abstract class Shape {
    class book {
        String name;
        String author;
        float price;
        int num-pages;
    }
}
```

```
book() {}  
book(String name, String  
author, float num-pages,  
float price) {  
    this.name = name;  
    this.author = author;  
    this.num-pages = num-pages;  
    this.price = price;  
}
```

```
void set-details() {
```

```
Scanner sc = new Scanner(System.in);
System.out.println("enter bookname, author,
price, num-pages");
name = sc.next();
author = sc.next();
```

```
price = sc.nextFloat();
num-pages = sc.nextInt();
}
```

```
void get-details(book b[], String search-name)
```

```
// int n = b.length;
```

```
// boolean found = false;
```

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

```
// if ( b[i].name.equals(search-name)) {
```

```
// found = true;
```

```
System.out.println("details of the
book is :");
```

```
System.out.println(b[i].toString());
```

Q. 4

```
// if (found == false) {
```

```

    // System.out.print("no book exists with
    // such name");
    // }
}

public String toString() {
    return "The book " + name + " was
    written by " + author + " it consists of "
    + numPages + " pages and costs around " + price;
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    // System.out.println("enter no. of books:");
    // int n = sc.nextInt();
    // book b[] = new book[n];
    // for (int i=0; i<n; i++) {
    //     b[i] = new book();
    //     b[i].setDetails();
    // }
    // b[i].getDetails(b, searchName);

    // System.out.println("enter name of book
    // to search");
    // String searchName = sc.next();
    // b[0].getDetails(b, searchName);
}

```

Output: Ob = SetBook{ }
Ob = GetBook{ }

```

book b[] = new book[3];
book b[0] = new book("Java", "S", 90, 200);
b[1] = new book();
b[2] = new book();
b[1].setDetails();
b[2].setDetails();
b[0].getDetails();
b[1].getDetails();
b[2].getDetails();

```

Algorithm: Step 1

Step 1: Create class book and initialize name, author, price, num-pages.

Step 2: Create method set-details() in that print bookname, author, price, num-pages. Get user input for these.

Step 3: Create method get-details(), pass parameters book b[], String print the details of the book call function toString().

Step 4: Create method toString()
 return "To book " + name + " was written by "
 + author + " it consists of " + num-pages +
 " pages and costs " + price;

Step 5: Create main function.

Get user input for no. of books.

Create book b[] array, and

Run a loop, call object book().

call function set-details().

Run a loop, and call get-details().

Step 6: Stop.

Output: enter bookname, author, price, num-pages

ABC XYZ 100 100

enter bookname, author, price, numpages

PQR LMN 200 200

the book JAVA was written by S it consists of

90 pages and costs 200.0 rupees

the book ABC was written by XYZ if consists

of 100 pages and costs 100.0 rupees

the book PQR was written by LMN it consists of 200 pages and costs 200.0 rupees.

Algorithm: Step 1: Start

Step 2: Create abstract class Shape()

Initialize x, y.

(call get-area() function)

Step 3: Create main function and then
Create 3 objects Circle, Rectangle,
Triangle under Shape parent class

Step 4: Create Circle class extending Shape
Create constructor Circle()

Get user input for radius value.

Create function void-area() and
print area of circle $3.14 * x * y$

Step 5: Create Rectangle Class extending Shape
Create constructor Rectangle().

Get user input for length, breadth.

Create function void-area() and
print area of rectangle $x * y$

Step 6: Create Triangle class extending Shape
Create constructor Triangle()

Get user input for base, height.

Create function void-area() and

print area of triangle $0.5 * x * y$.

Step 7: Stop.

10/12/21

Page No. 21
Date: 11/12/21

④ import java.util.Scanner;
abstract class Shape {
 int x, y;
 abstract void area();
}

class Circle extends Shape {
 Circle() {

Scanner sc = new Scanner(System.in);
 System.out.println("enter radius:");
 x = sc.nextInt();

y = x;

}
 void area() {

System.out.println("area of circle = " + 3.14 * x * y);
 }

class Rectangle extends Shape {

Rectangle() {

Scanner sc = new Scanner(System.in);

System.out.println("enter length, breadth");

x = sc.nextInt();

y = sc.nextInt();
 }

void area() {

System.out.println("area of rectangle = " + x * y);
 }

class Triangle extends Shape {

Triangle() {

Scanner sc = new Scanner(System.in);

System.out.println("enter base, height");

x = sc.nextInt();

y = sc.nextInt();

```
void area() {
```

```
    System.out.println("Area of triangle = "+0.5*x*x);
```

```
}
```

```
class Area1 {
```

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

```
        //System.out.println("Abubakar 13M22CS010");
```

```
        Circle obj1 = new Circle();
```

```
        obj1.area();
```

```
        Rectangle obj2 = new Rectangle();
```

```
        obj2.area();
```

```
        Triangle obj3 = new Triangle();
```

```
        obj3.area();
```

```
}
```

Output: Enter radius : 1

Area of circle = 3.14

Enter length, breadth : 2 2

Area of rectangle = 4

Enter base, height : 2 2

Area of triangle = 2.0

O/P from
sum

19/11/23

Q3) Develop program to create class Bank that maintains 2 account for current, savings.
savings → compound interest
current → min bal.

Sol 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 No.: " + accno);

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

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

class CurAcct extends Account {

double minBalance;

double serviceCharge;

public CurAcct (String customerName, long accno)

super (customerName, accno, "Current");

this.minBalance = 500.0;

this.serviceCharge = 50.0;

public void withdraw (double amount) {

if (balance - amount >= minBalance)

balance - amount;

Cheque issued

System.out.println ("Withdrawal Successful");
Current balance : Rs. " + balance);

else {

System.out.println ("WithInsufficient funds.");

}

public void imposeServiceCharge () {
if (balance < minBalance) {
balance += serviceCharge;

System.out.println ("Service charge imposed.");

Current Balance : Re " + balance);

}

class SavAcct extends Account {

double interestRate;

public SavAccount (String customerName, long accno)
super (customerName, accno, "Savings");
this.interestRate = 0.05;

public void compoundDepositInterest () {
double initialAmount, int term;

double interest = balance *

double compoundInterest = initialAmount *

Math. pow ((1 + interestRate), term) -
initialAmount;

balance += compoundInterest;

}

public class Bank {

public static void main (String [] args) {

Scanner scanner = new Scanner (System.in);

System.out.println ("Choose acc type:");

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

Enter 1 or 2 ");

```
Scanner. nextInt();  
System.out.println ("Enter customer name :");  
Scanner.next();  
System.out.println ("Enter accno :");  
Scanner.nextInt();  
if (choice == 1) {  
    curAcct = new CurAcct (customerName, accNo);  
    System.out.println ("Enter initial balance : Rs");  
    double initialBalance = Scanner.nextDouble();  
    curAcct.balance = initialBalance;  
    System.out.println ("Enter withdrawal amount : Rs");  
    double withdrawalAmount = Scanner.nextDouble();  
    curAcct.withdraw (withdrawalAmount);  
    curAcct.displayBalance ();  
    curAcct.serviceCharge ();  
}  
else if (choice == 2) {  
    SavAcct savAcct = new SavAcct (customerName, accNo);  
    System.out.println ("Enter initial Balance :");  
    double initialBalance = scanner.nextDouble();  
    System.out.println ("Enter withdrawal amount :");  
    double withdrawalAmount = scanner.nextDouble();  
    SavAcct.balance = withdrawalAmount;  
    System.out.println ("Withdrawal Successful ,  
    Current Balance = Rs " + SavAcct.Balance);  
    System.out.println ("Enter interest rate :");  
    double interestRate = scanner.nextDouble();  
    savAcct.interestRate = interestRate;  
    savAcct.displayBalance ();  
    System.out.println ("Enter term (in years)  
    for C.I. calculation : ");
```

int term = scanner.nextInt();

SavAcct compoundInterest (initialBalance, term);
SavAccount.displayBalance();

} else {

System.out.println ("Invalid choice");

Algorithm

Step 1: Start main() function

Step 2: Create class Account.

Initialize customerName, accno, accountType,
balance.

Step 3: Create constructor with all these parameters.

Step 4: Create method displayBalance to
display all these values.

Step 5: Create class curAcct extending Account

Initialize minBalance, serviceCharge.

Step 6: Create constructor.

Take initialization from super class

and give values to minBalance & service
charge.

Step 7: Create method withdraw.

If balance - amount ≥ 0 , then do it

else print insufficient funds.

Create method

Step 8: Create method imposeServiceCharge().

If balance $<$ minBalance, balance = Service
charge.

Step 9: Create class SavAcct extending Account.

Initialize interestRate.

Step 10: Create constructor SavAccount and

initialize from super class.

Give value to interestRate.

Step 11: Create method compoundInterest to calculate mathematical compound Interest and add to balance.

Step 12: Create class Bank.

Create main function.

Print Choose account type
1. Current
2. Savings.

Take user input for choice, name, accno.

Step 13: If choice = 1, create object of CurAct.
Initialize initialBalance, withdrawal amount.

(Call function methods withdraw(),
displayBalance(), serviceCharge());

~~Step 14~~ Or else choice = 2, Create obj of SavAct.
Initialize initialBalance, withdrawal amount,
interest rate.

call methods withdraw(), compoundInterest(),
displayBalance();

Else print invalid choice.

Step 14: End Step.

~~Final
part~~

Output: Choose account type : 1. Current 2. Savings
Enter choice : 1

Enter customer name : Abubakar

Enter account number : 1248999

Enter initial balance : \$400

Enter withdrawal amount : \$100

Insufficient funds. Withdrawal not allowed.

Service Charge imposed. Current Balance : \$350

Account Number : 3214567

Account Type: Current

Balance : \$350

Output 2: Choose Account type

1. Current
2. Savings

Enter choice: 2

Enter customer name: Abubakar

Enter account number: 12345678

Enter initial balance: \$ 8000

Enter withdrawal amount: \$ 1500.

Withdrawal successful. Current Balance: \$ 6500.

Enter interest rate: 0.08

Account Number: 12345678

Customer Name: Abubakar

Account type: Savings

Balance: \$ 6500

Enter term (in year) for CI calculation: 3

Compound Interest depositer. Current

Balance: \$ 18577.696

Account Number: 12345678

Customer Name: Abubakar

Account type: Savings

Balance: \$ 18577.696

Q91

2/2/24

6 →

Packages: Create packages and import & display CIE, SEE

package CIE;

import java.util.*;

public class Student {

 public int sem;

 public String usn;

 public String name;

 public void accept() {

 Scanner sc = new Scanner(System.in);

 System.out.println("Enter usn, name, sem");

 usn = sc.nextLine();

 name = sc.nextLine();

 sem = sc.nextLine();

}

package SEE;

public class Internals {

 public int in_marks[] = new int[5];

}

package SEE;

import CIE.Student;

public class External extends Student {

 public int ex_marks[] = new int[5];

}

import

import java.util.*;

import SEE.*;

import CIE.*;

public class FinalMarks {

 public static void main (String [] args) {

```
int f-marks[] = 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.Internal s[] = new CIE.Internal[n];
for (int i=0; i<n; i++) {
    st[i] = new SEE.External();
    s[i] = new CIE.Internal();
    System.out.println("Enter details " + (i+1));
    st[i].accept();
    for (int j=0; j<5; j++) {
        System.out.println("Enter internal
and external marks of subject " + (i+1));
        st[i].in-marks[j] = sc.nextInt();
        st[i].ex-marks[j] = sc.nextInt();
        f-marks[j] = s[i].in-marks[j] +
                      + st[i].ex-marks[j];
    }
}
```

~~System.out.println("Final marks of"
+ st[i].name);~~

~~for (int k=0; k<5; k++) {~~

~~System.out.println("Course " + (k+1) +
" = " + f-marks[k]);~~

O/P: Enter no. of students : 2
Enter USN, Name, Semester :

1BN22CS610

Abubakar

3.

Enter internal and external marks of subject 1: 49 48
 Enter internal and external marks of subject 2: 47 49
 Enter internal and external marks of subject 3: 50 49
 Enter internal and external marks of subject 4: 48 46
 Enter internal and external marks of subject 5: 45 50
 Final marks of Abubakar:

Course 1 = 97

Course 2 = 96

Course 3 = 99

Course 4 = 94

Course 5 = 95

Algorithm:

Step1: Start

Step2: Create a package named CIE

Create a public class named Student

Initialize public variables usn, name, sem.

Step3: Create a public method named accept()

Print "Enter usn, name, sem:"

Take user input

Step4: Create a public method class named Internals.

Initialize public array in-marks.

Step5: Create a package SEE and import

CIE.Student. from package CIE

Create public class External, extending Student

Initialize public array ex-marks.

Step6: Import SEE.* and CIE.*

Create a public class named finalMarks.

Create main method.

Initialize array f-marks.

Take user input for no. of students.

Step 7:

Create a st[] obj for SEF, external and a s[] obj for CIE Internals and new arrays.

Step 8: Run loop from 0 to n

Accept s[] details.

Run loop from 0 to 5

Take in-marks & ex-marks from user and take their sum in fmarks. Print it.

Step 9: Run loop from 0 to 5

Print all the details.

Step 10: Stop.

~~Q2/2M~~

(7) (Q8) Exception Handling.

```
import java.util.Scanner;
class WrongAge extends Exception {
    String message;
    super(message);
    public WrongAge (int fatherAge) throws WrongAge {
        message = "Age cannot be negative";
    }
}
```

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

```
}
```

```
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 sc = new Scanner (System.in);
        System.out.println ("Enter father's &
                            son's age : ");
        int fa = sc.nextInt ();
        int sa = sc.nextInt ();
        try {
            Son s = new Son (fa, sa);
            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 ());
        }
    }
}

```

off: Enter father's age & son's age:
 30
 31

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

Step 1: Start
Algorithm: Step 2: create a class named WrongAge which is extending Exception.
 Create constructor and call super(message).

Step 2: Create a class named Father.
 Initialize fatherAge.
 Create constructor that throws WrongAge if fatherAge < 0.

Step 3: Create a class named Son which is extending Father.
 Initialize sonAge.

Create constructor that throws WrongAge exception if sonAge <= fatherAge.

Step 4: Create a main class FatherSon.

In main function,

print "Enter father's & son's details."

Take user input for the two, father & son.

Inside try block, create object of Son s, put father, son as parameters.

Print father's & son's age.

Inside catch block print error & e.getMessage().

Step 5: Stop.

16/12/24

Q) Multi Threading

```
class A extends Thread {  
    int t1, time;  
    A() {  
        t1 = 0;  
        time = 25000; // Circled  
    }  
    public void run() {  
        while (t1 <= time) {  
            System.out.println("BMS College  
of Engineering");  
            try {  
                sleep(10000);  
            } catch (Exception e) {  
                System.out.println("Error");  
            } t1 += 10000;  
        }  
    }  
}
```

```
class B extends Thread {  
    int t2, time;  
    B() {  
        t2 = 0;  
        time = 25000;  
    }  
    public void run() {  
        while (t2 <= time) {  
            System.out.println("CSE");  
            try {  
                sleep(2000);  
            } catch (Exception e) {  
                System.out.println("Error");  
            } t2 += 2000;  
        }  
    }  
}
```

88

}

catch (Exception e) {

System.out.println("Error");

} t2 += 2000;

}

}

}

class MThread {

public static void main () {

A a = new A();

B b = new B();

a.start();

b.start();

}

}

o/p: BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

Algorithm: Step 1: Start

Step 2: Create a class A which extends Thread.

Initialize t₁, time inside its constructor.

Override run method.

while t₁ ≤ time,

print BMS College of Engineering -

Inside try block, ~~block the~~ ^{block the} thread

for 10 seconds using sleep(10000)

Catch exception e and

display error.

Increment t₁ by 10000 every iteration.

Step 3: Create a class B which extends Thread.

Initialize t₂, time inside its constructor.

Override the run method.

while t₂ ≤ time,

print CSE.

Inside try block, block the thread for 2 seconds using sleep(2000).

Catch exception e & display error.

Increment t₂ by 2000 every iteration.

Step 4: Create public class Thread.

Inside main function, create objects for class A & B respectively.

Call their respective run methods using a.start() & b.start().

Step 5: Stop.

16.02.24

✓ ✓