

B.M.S. COLLEGE OF ENGINEERING
Basavanagudi, Bengaluru- 560019
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



LAB REPORT

On

Object Oriented Java Programming
(23CS3PCOOJ)

Submitted By :

ABUBAKAR MOHAMMEDSHAFEE MATTE
1BM22CS010

In partial fulfilment of
BACHELOR OF ENGINEERING
In
COMPUTER SCIENCE AND ENGINEERING
2023-24

Faculty-In-Charge
Swathi Sridharan
Assistant Professor
Department of Computer Science and Engineering

SL. NO	DATE	TOPIC	PageNo
1	08/12/23	Sample Programs	3
2	22/12/23	Quadratic Equation	13
3	29/12/23	Student SGPA Calculation	18
4	12/01/24	Book Details	23
5	12/01/24	Area Calculation	28
6	19/01/24	Bank Account Details	33
7	02/02/24	External And Internal Marks	41
8	16/02/24	Exception Handling	47
8	16/02/24	Threads	52
9	23/02/24	AWT	57

LAB-1
Sample Programs

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 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 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);
    }
}
```

3.

O/p:

Quotient is : 7

Remainder is : 1

5) Multiply 2 no.s.

```
import java.util.Scanner;
public class multi {
    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);
    }
}
```

3

O/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);  
    }  
}
```

Op: --Before Swap--
First no. = 1.20 Second no. = 2.45 --After swap-- First no. = 2.45 Second no.
= 1.20

① Algorithm:

Step1: Start

Step2: Create a reader instance to take input

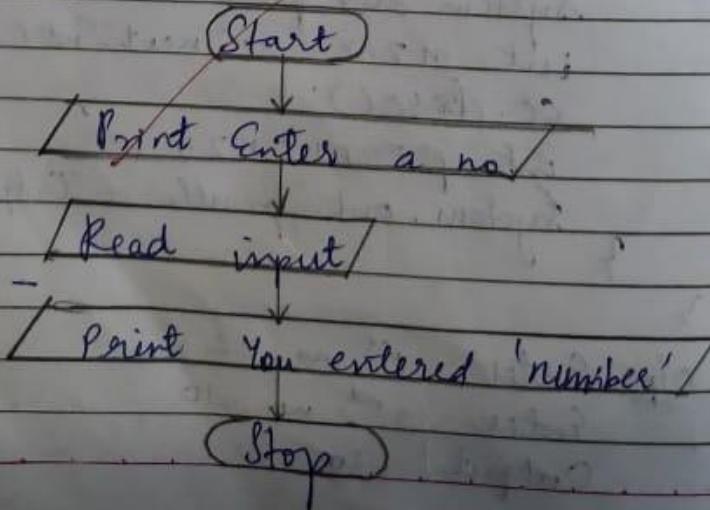
Step3: Print Enter no.

Step4: Read no. obtained by user

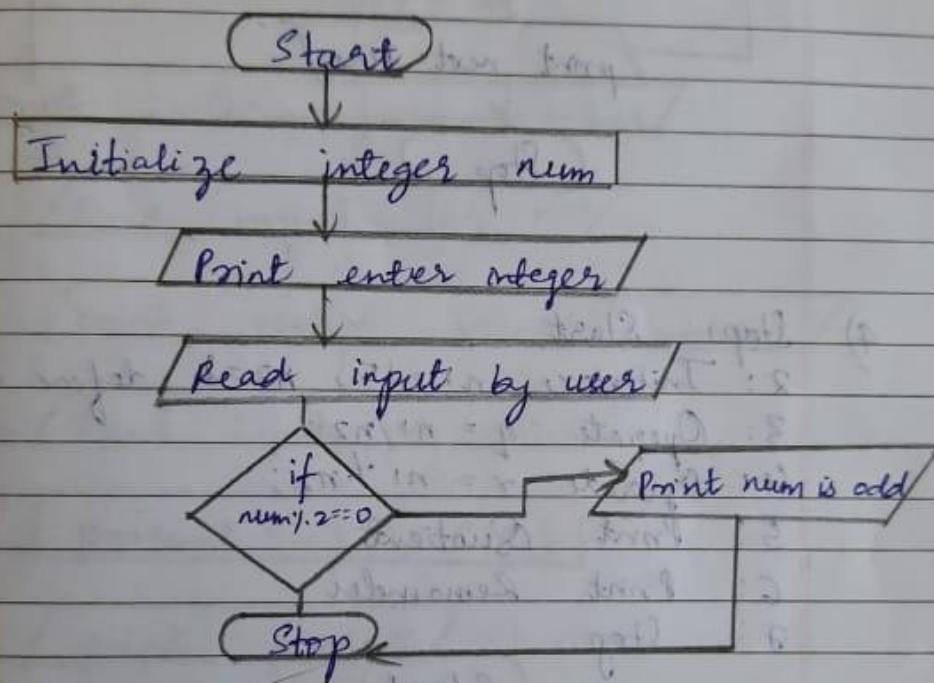
Step5: Print you entered 'number'

Step6: 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 number 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 r, c, n & initialize $n = 8$

~~for ($r = 0, r < n, r++$)~~

~~for ($c = 0, c < r, c++$)~~

~~print '*'~~

~~print next line~~

Stop

7) 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 $n_1 = 15$ & $n_2 = 2$

$q = n_1 / n_2$

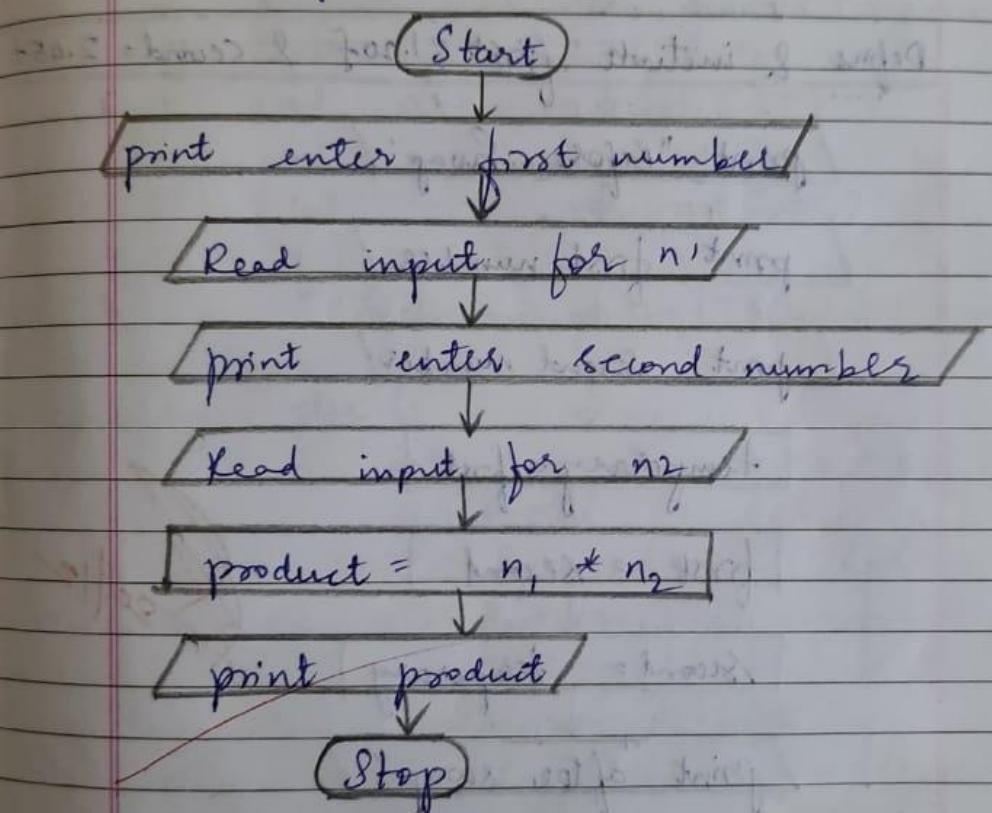
$r = n_1 \% n_2$

Print Quotient

Print Remainder

Stop

- 5) Step 1: 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 = n1 * n2
- 7: print product
- 8: Stop



- 6) Step 1: 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

```
C:\Users\bmsce\Desktop\1BM22CS010>javac HelloWorld.java  
C:\Users\bmsce\Desktop\1BM22CS010>java HelloWorld  
Abubakar M Matte 1BM22CS010  
Enter a number:  
3  
You entered: 3
```

```
C:\Users\bmsce\Desktop\1BM22CS010>javac EvenOdd.java  
C:\Users\bmsce\Desktop\1BM22CS010>java EvenOdd  
Abubakar M Matte 1BM22CS010  
Enter an integer number:  
6  
6 is an even number
```

```
C:\Users\bmsce\Desktop\1BM22CS010>javac EvenOdd.java  
C:\Users\bmsce\Desktop\1BM22CS010>java EvenOdd  
Abubakar M Matte 1BM22CS010  
Enter an integer number:  
6  
6 is an even number
```

```
C:\Users\bmsce\Desktop\1BM22CS010>javac quotientRemainder.java  
C:\Users\bmsce\Desktop\1BM22CS010>java quotientRemainder  
Abubakar M Matte 1BM22CS010  
Quotient is: 7  
Remainder is: 1
```

```
C:\Users\bmsce\Desktop\1BM22CS010>javac quotientRemainder.java  
C:\Users\bmsce\Desktop\1BM22CS010>java quotientRemainder  
Abubakar M Matte 1BM22CS010  
Quotient is: 7  
Remainder is: 1
```

```
C:\Users\bmsce\Desktop\1BM22CS010>javac quotientRemainder.java  
C:\Users\bmsce\Desktop\1BM22CS010>java quotientRemainder  
Abubakar M Matte 1BM22CS010  
Quotient is: 7  
Remainder is: 1
```

LAB-2

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

22/12/23

66
Page No. 9
Date 1/1/23

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);

}

y

3

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

Step 4: $d = b^2 - 4ac$

Step 5: if $d > 0$

print "roots are real"

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

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

print (r_1, r_2) goto Step 8.

Step 6: if $d < 0$

print ("Roots are imaginary")

There are no real solutions

$$r_1 = \frac{-b}{(2a)}$$

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

print ($r_1 + i r_2$)

print ($r_1 - i r_2$)

goto step 8.

Step 7: if $d=0$

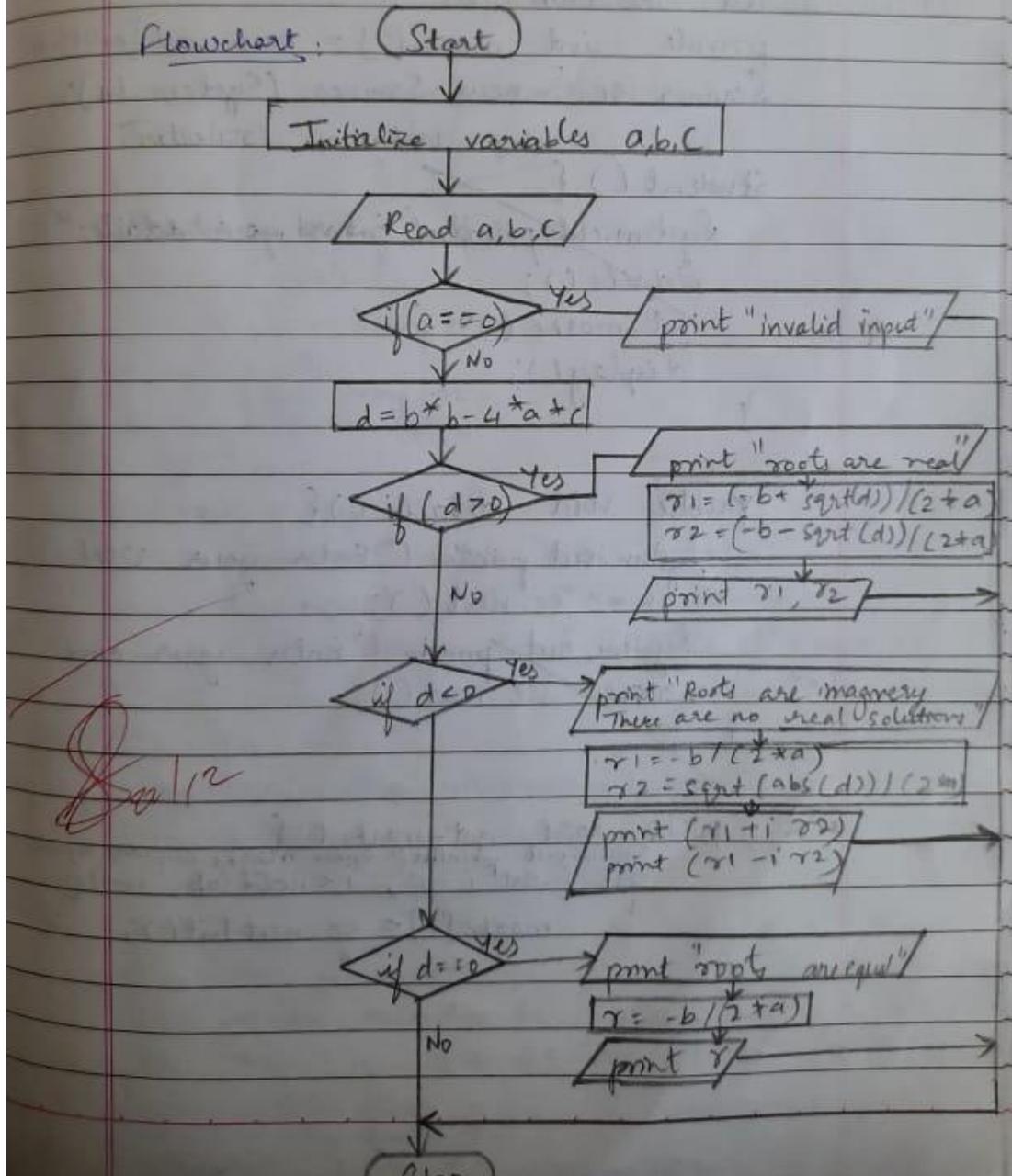
print " Roots are equal"

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

print r

Step 8: Stop.

Flowchart: (Start)



```
C:\Users\bmsce\Desktop\1BM22CS010>java quad
Name: Abubakar Mohammedshafee Matte
USN: 1BM22CS010
Enter coefficients:
0 1 1
Invalid Input.

C:\Users\bmsce\Desktop\1BM22CS010>javac quad.java

C:\Users\bmsce\Desktop\1BM22CS010>java quad
Name: Abubakar Mohammedshafee Matte
USN: 1BM22CS010
Enter coefficients:
1 2 1
Roots are equal:
-1.0

C:\Users\bmsce\Desktop\1BM22CS010>javac quad.java

C:\Users\bmsce\Desktop\1BM22CS010>java quad
Name: Abubakar Mohammedshafee Matte
USN: 1BM22CS010
Enter coefficients:
1 -6 5
Roots are real:
5.0
1.0

C:\Users\bmsce\Desktop\1BM22CS010>javac quad.java

C:\Users\bmsce\Desktop\1BM22CS010>java quad
Name: Abubakar Mohammedshafee Matte
USN: 1BM22CS010
Enter coefficients:
5 1 5
Roots are imaginary. There are no real solutions. Complex solutions:
-0.1+i0.99498737
-0.1-i0.99498737
```

LAB-3

/*Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student. */

29/12/23

SGPA Calculation.

```
import java.util.Scanner;  
public class Student {  
    String usn;  
    String name;  
    private static int credits[] = {4, 4, 4, 3, 2};  
    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 < noofSub; 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("VSN :" + usn);
    System.out.println("Name :" + name);
    System.out.println("SGPA is :" + sgpa());
}

```

```

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

```

Op: Enter your details :

Enter marks according to credits:

99 94 92 87 83 96 83 90

VSN: 1BM22CS010

Name: Alubakar

CGPA is : 9.10 18 18 18 18 18 18 2

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(), getMarks(), display()

Step 5 Create method details()

print (Enter usn) print (Enter name)

Take user input for USN and name

Step 6 Create method getDetails()

print (Enter your marks)

Take user input for marks.

Step 7 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, $\text{sgpa} = \frac{\text{temp} + \text{credit} * (\text{marks})}{10}$

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

else temp+=0

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

Step 8 Create method display()

print USN, name and call sgpa()

Step 9 Create main function

and make object s1 of Student.

Step 10 Stop

29/12/23

```
C:\Users\bmsce\Desktop\1BM22CS010>javac Student.java
C:\Users\bmsce\Desktop\1BM22CS010>java Student
Enter your details:
Abubakar Mohammedshafee Matte 1BM22CS010
Enter your USN:
1BM22CS010
Enter your name:
Abubakar
Enter your marks accoring to credits-4,4,4,3,3,2,1,1:
99
94
92
87
83
96
83
90
USN: 1BM22CS010
Name: Abubakar
SGPA is: 9.681818181818182
```

LAB-4

/*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 objects. Include a toString() method that could display the
complete details of the book. Develop a Java program to create n book objects.*/

12/1/24

66
Page No. 18
Date: 55

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

Sol:

```
import java.util.Scanner;  
abstract class Shape {  
    class book {  
        String name;  
        String author;  
        float price;  
        int numPages;  
        book() {}  
        book(String name, String  
              author, float numPages,  
              float price) {  
            this.name = name;  
            this.author = author;  
            this.numPages = numPages;  
            this.price = price;  
        }  
    }  
}
```

```
void setDetails() {  
    Scanner sc = new Scanner(System.in);  
    System.out.println("enter bookname, author,  
                      price, num-pages");  
    name = sc.next();  
    author = sc.next();  
    price = sc.nextFloat();  
    numPages = sc.nextInt();  
}
```

```
void getDetails(book b[], String searchName){  
    // int n = b.length;  
    // boolean found = false;  
    // for (int i=0; i<n; i++) {  
    //     if (b[i].name.equals(searchName)) {  
    //         found = true;  
    //         System.out.println("details of the  
    // book is :");  
    //         System.out.println(b[i].toString());  
    //     }  
    // }  
    // if (found == false) {  
    // }
```

24

```

// System.out.println("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();
    }

    // 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", 9, 20);
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[], Stone. print the details of the book call function toString().

Step 4: Create method toString() return "The 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 it 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.

```
C:\Users\bmsce\Desktop\1BM22CS010>javac book.java
C:\Users\bmsce\Desktop\1BM22CS010>java book
Abubakar Mohammedshafee Matte 1BM22CS010
enter bookname,author,price,num_pages
ABC
XYZ
100
100
enter bookname,author,price,num_pages
PQR
LMN
200
200
the book Java was written by Strange it consists of 9857 pages and costs 243.0rupees.
the book ABC was written by XYZ it consists of 100 pages and costs 100.0rupees.
the book PQR was written by LMN it consists of 200 pages and costs 200.0rupees.
```

LAB-5

/*Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named 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*/

Date: 10/12/24

Page No. 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 * 2 * r);  
}
```

```
class Area1 {  
    public static void main(String[] args) {  
        // System.out.println("Abubakar IX M2215016");  
        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

Off now
Ali

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.

```
C:\Users\bmsce\Desktop\1BM22CS010>java Area
Abubakar 1BM22CS010
enter the radius of the circle
1
area of circle is 3.14
enter the length and breadth of the rectangle
2
2
area of rectangle is 4
enter the base and height of the triangle
2
2
area of triangle is 2.0
```

LAB-6

/*Develop a Java 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 a minimum balance and if the 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:

- 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 the balance. */

Q8) Develop program to create class Bank that maintains 2 accounts for current, savings.
 savings → compound interest
 current → min balance

```
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 = 1500.0;
        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 ("With insufficient funds.");

}

public void imposeServiceCharge () {

if (balance < minBalance) {

balance -= serviceCharge;

System.out.println ("Service charge imposed.

Current Balance : Rs. " + balance);

}

class SavAcc extends Account {

double interestRate;

public SavAccount (String customerName, long accno)

super (customerName, accno, "Savings");

this. interestRate = 0.05;

public void compoundDepositInterest () {

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.nextLong();
if (choice == 1) {
    curAcct = new CurAcct(customerName, accno);
    System.out.println("Enter initial balance: ");
    double initialBalance = scanner.nextDouble();
    curAcct.balance = initialBalance;
    System.out.println("Enter withdrawal amount: ");
    double withdrawalAmount = scanner.nextDouble();
    curAcct.withdraw(withdrawalAmount);
    curAcct.displayBalance();
    curAcct.serviceCharge();
}

else if (choice == 2) {
    SavAcct savAccount = 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();
    savAccount.interestRate = interestRate;
    savAccount.displayBalance();
    System.out.println("Enter term (in years) for CI calculation: ");
}

```

```
int term = Scanner.nextInt();  
SavAcct compoundInterest (floatBalance, term);  
SavAccount.displayBalance();  
}  
else {  
    System.out.println ("Invalid choice");  
}  
}  
}  
}
```

Algorithm

Step 1: Start

Step 2: Create class Account

Initialize CustomerName, accno, accountType, balance

Step 3: Create constructor with all these parameters

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

Step 5: Create class curAcct extending Account

Initialize minBalance, serviceCharge

Step 6: Create ~~for~~ constructor

Take initialization from super class

and give values to minBalance & serviceCharge

Step 7: Create method withdraw

If balance - amount \geq 0, then do it

Otherwise, print insufficient funds.

Create method

Step 8: Create method imposeServiceCharge()

If balance < minBalance, balance = $balance - serviceCharge$

Step 9: Create class SavAcct extending Account

Initialize interestRate

Step 10: Create constructor SavAccount and

initialize from super class

Given 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 acc type
1. Current
2. Savings.

Take user input for choice, name, accno

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

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

Step 14: On else choice = 2, create obj of SavAcct.
Initialize initialBalance, withdrawal amount,
interest rate.

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

Else print invalid choice

Step 14: End Step

~~From
Part~~

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

Enter customer name : Abubakar

Enter account number : 1248991

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

End of program

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 deposited. Current

Balance: \$ 8577.696

Account Number: 12345678

Customer Name: Abubakar

Account type: Savings

Balance: \$ 8577.696

8/11

```
C:\Users\bmsce\Desktop\1BM22CS010>javac Bank.java

C:\Users\bmsce\Desktop\1BM22CS010>java Bank
Abubakar Matte 1BM22CS010
Choose account type:
1. Current
2. Savings
Enter choice (1 or 2): 1
Enter customer name: Abubakar
Enter account number: 3625245
Enter initial balance: $400
Enter withdrawal amount: $100
Insufficient funds. Withdrawal not allowed.
Service charge imposed. Current Balance: Rs.350.0
Account Number: 3625245
Customer Name: Abubakar
Account Type: Current
Balance: $350.0

C:\Users\bmsce\Desktop\1BM22CS010>javac Bank.java

C:\Users\bmsce\Desktop\1BM22CS010>java Bank
Abubakar Matte 1BM22CS010
Choose account type:
1. Current
2. Savings
Enter choice (1 or 2): 1
Enter customer name: Abubakar
Enter account number: 9857932
Enter initial balance: $10000
Enter withdrawal amount: $1000
Withdrawal successful. Current Balance: $9000.0
Account Number: 9857932
Customer Name: Abubakar
Account Type: Current
Balance: $9000.0

C:\Users\bmsce\Desktop\1BM22CS010>javac Bank.java

C:\Users\bmsce\Desktop\1BM22CS010>java Bank
Abubakar Matte 1BM22CS010
Choose account type:
1. Current
2. Savings
Enter choice (1 or 2): 2
Enter customer name: Abubakar
Enter account number: 12345678
Enter initial balance: $8000
Enter withdrawal amount: $1500
Withdrawal successful. Current Balance: $6500.0
Enter interest rate: 0.08
Account Number: 12345678
Customer Name: Abubakar
Account Type: Savings
Balance: $6500.0
Enter term (in years) for compound interest calculation: 3
Compound Interest deposited. Current Balance: Rs.8577.696000000002
Account Number: 12345678
Customer Name: Abubakar
Account Type: Savings
Balance: $8577.696000000002
```

LAB-7

/*Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five 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 five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.*/

2/2/24

6 →

Packages: Create packages and import & display.

CIE SEE

99

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();

}

3

package CIE;

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));
    s[i].accept();
    for (int j=0; j<5; j++) {
        System.out.println("Enter internal
and external marks of subject " + (i+1));
        s[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:
 1BN22CS010
 Abubakar

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 CIF
 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 Internals .
 Initialize public array in-marks.

Step5: Create a package SEE and import
 CIF.Student from package CIF

Create public class External which is
 named extending Student

Initialize public array ex-marks

Step6: Import SEE.* and CIF.*.

Create a public class 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 CIF, 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.

~~(8/2) / 21~~

```
C:\Users\bmsce\Desktop\1BM22CS010\CIE_SEE>javac -d . FinalMarks.java
C:\Users\bmsce\Desktop\1BM22CS010\CIE_SEE>java FinalMarks
Enter n:
2
Enter details 1
Enter USN, Name, Semester:
1BM22CS010
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
49
Final marks of Abubakar
Course 1 = 97
Course 2 = 96
Course 3 = 99
Course 4 = 94
Course 5 = 94
Enter details 2
Enter USN, Name, Semester:
1BM22CS010
Abubakar
4
Enter internal and external marks of subject 1
50
50
Enter internal and external marks of subject 2
49
48
Enter internal and external marks of subject 3
47
49
Enter internal and external marks of subject 4
48
48
Enter internal and external marks of subject 5
49
50
Final marks of Abubakar
Course 1 = 100
Course 2 = 97
Course 3 = 96
Course 4 = 96
Course 5 = 99
```

LAB-8

/*Write a program that demonstrates handling of exceptions in inheritance tree.
Create a base class called “Father” and derived class called “Son” which extends the
base class. In Father class, implement a constructor which takes the age and throws
the exception WrongAge() when the input age<0. In Son class, implement a
constructor that cases both father and son’s age and throws an exception if son’s age is
>=father,s age */

16/2/24
Q18)

66
Page No. 33
Date: 1/1/99

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

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

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

```
public class fatherson {  
    public static void main (String [] args) {  
        Scanner 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());  
        }  
    }  
}
```

Output:
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 1: 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, fa & sa.

Inside try block, create object of Son s, put fa, sa as parameters.

Print father's & son's age.

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

Step 5: Stop.

```
C:\Users\bmsce>cd C:\Users\bmsce\Desktop\1BM22CS010

C:\Users\bmsce\Desktop\1BM22CS010>javac fatherson.java

C:\Users\bmsce\Desktop\1BM22CS010>java fatherson
Name: Abubakar Mohammedshafee Matte
USN: 1BM22CS010
Enter father's age and son's age:
45
15
Father's age: 45
Son's age: 15

C:\Users\bmsce\Desktop\1BM22CS010>javac fatherson.java

C:\Users\bmsce\Desktop\1BM22CS010>java fatherson
Name: Abubakar Mohammedshafee Matte
USN: 1BM22CS010
Enter father's age and son's age:
30
31
Error: Son's age must be less than Father's age

C:\Users\bmsce\Desktop\1BM22CS010>javac fatherson.java

C:\Users\bmsce\Desktop\1BM22CS010>java fatherson
Name: Abubakar Mohammedshafee Matte
USN: 1BM22CS010
Enter father's age and son's age:
-1
-46735
Error: Age cannot be negative
```

LAB-8

```
/*Write a program which creates two threads, one thread displaying "BMS College of  
Engineering" once every ten seconds and another displaying "CSE" once every two seconds
```

16/2/24
7) (8) Multi Threading

class A extends Thread {

int t1, time;

A() {

t1 = 0;

time = 25000;

}

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;  
}  
}
```

```
class MThread {  
    public static void main () {  
        A a = new A();  
        B b = new B();  
        a.start();  
        b.start();  
    }  
}
```

off: BMS College of Engineering

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(1000)

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 CSF.

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

Catch exception e and display error.

Increment t2 by 2000 every iteration

Step 4: Create public class MThread.

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

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

Step 5: Stop.

16.02.24

Pray

```
C:\Users\bmsce\Desktop\1BM22CS010>javac MThread.java

C:\Users\bmsce\Desktop\1BM22CS010>java MThread
Name: Abubakar Mohammedshafee Matte
USN: 1BM22CS010
BMS COLLEGE OF ENGINEERING
CSE
CSE
CSE
CSE
CSE
BMS COLLEGE OF ENGINEERING
CSE
CSE
CSE
CSE
CSE
BMS COLLEGE OF ENGINEERING
CSE
```

LAB-9

/*Java awt class*/

AWT

Page No. 39
Date: 11/11/2023

- java.awt: Provides interfaces that enable development of API methods that can be used to work with any Java runtime environment.
- JFrame: class, has decorations like border, title, supports button components.
- FlowLayout(): Arranges components in a left-to-right flow, much like para.
- setDefaultCloseOperation(): hides JFrame when user closes window.
- EXIT_ON_CLOSE: Exits application using System.exit method.
- JLabel: class, lets you display information on a JFrame.
- JTextField: lightweight component that allows editing of single line of text.
- JButton: class used to generate push-button control that triggers an ActionEvent when pressed.
- add() - adds frame
- ActionListeners - define what should be done when user performs certain operations.

- ActionEvent : Handles action performed event by invoking methods.
- parseInt() : static method that takes a string as argument & returns Integer.
- setText() : substitutes characters ^{works for} text for text in text field. First line only
- setVisible() : If true, it is visible on screen.
- Swing Utilities.invokeLater (new Runnable (){
Creates an instance of anonymous implementation of Runnable interface & passes it to invokeLater, which puts it to queue.
invokeLater performs asynchronously in AWT Event dispatcher thread.
- getText() : returns text from single-line text field.

```

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

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

        JLabel jlab = new JLabel("Enter
        divider and dividend:");
        JTextField aJtf = new JTextField(8);
        JTextField bJtf = new JTextField(8);
        JButton button = new JButton("Calculate");
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anolab = new JLabel();

        jfrm.add(err);
        jfrm.add(jlab);
        jfrm.add(aJtf);
        jfrm.add(bJtf);
        jfrm.add(button);
        jfrm.add(alab);
        jfrm.add(blab);
        jfrm.add(anolab);
    }
}

```

```

ActionListener l = new ActionListener() {
    public void actionPerformed (ActionEvent evt) {
        System.out.println ("Action event

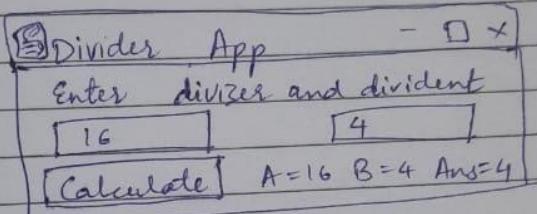
```

```
from a text field");  
});  
ajtf.addActionListener(l);  
bjtf.addActionListener(l);  
  
button.addActionListener(new ActionListener(){  
    public void actionPerformed(ActionEvent e)  
    { try {  
        int a = Integer.parseInt(ajtf.getText());  
        int b = Integer.parseInt(bjtf.getText());  
        int ans = a/b;  
  
        alab.setText("In A = " + a);  
        blab.setText("In B = " + b);  
        anslab.setText("In Ans = " + ans);  
    } catch (NumberFormatException e) {  
        alab.setText("");  
        blab.setText("");  
        anslab.setText("");  
        err.setText("Enter only integers!");  
    } catch (ArithmaticException e) {  
        alab.setText("");  
        blab.setText("");  
        anslab.setText("");  
        err.setText("B should be NON zero!");  
    }  
}};  
item.setVisible(true);  
});
```

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

```
getText());  
getText());
```

O/p:



me);

8/9/2022

egers!")

