

HRUTHVIK

MALLIKARJUNAPPA(1BM22C S114)

Q1. 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.

```
import java.util.Scanner;
import java.lang.Math;
class quadratic
{public static void main(String XX[])
{
    int a,b,c;
    System.out.println("enter the values of a,b,c respectively\n");
    Scanner s1= new Scanner(System.in);
    a = s1.nextInt();
    b = s1.nextInt();
    c = s1.nextInt();
    double d= b*b - 4*a*c ;
    System.out.println("a = " + a + " b = " + b + " c = " + c);
    if(a==0) {System.out.println("not a quadratic equation");}
    else if( d>0)
    {
        System.out.println("the equation has two real and different solutions");
        double r1=(-b + Math.sqrt(d))/(2*a);
        double r2=(-b - Math.sqrt(d))/(2*a);
        System.out.println("r1 = " + r1);
        System.out.println("r2 = " + r2);
    }

    else if(d==0)
    {
        System.out.println("the equation has real and equal solutions");
        double r1= -b/(2*a);
        double r2= -b/(2*a);
        System.out.println("r1 = " + r1);
        System.out.println("r2 = " + r2);
    }

    else if(d<0)
    {
```

```

        System.out.println("the equation has unreal solutions");
    }
}

```

LAB-I

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

```

class Quadratic
{
    public static void main(String[] args)
    {
        int a, b, c;
        double x, y;
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the values of a, b and c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
        int k = (b*b) - 4*a*c;
        if (k < 0)
        {
            System.out.println("No real roots");
        }
        else
        {
            double l = Math.sqrt(k);
            x = (-b - l) / 2*a;
            y = (-b + l) / 2*a;
            System.out.println("Roots of given equations"
                               + "\nx = " + x + " and y = " + y);
        }
    }
}

```

OUTPUT

Enter coefficients -

1

-8

15

Real roots are 50 and 30

Enter coefficients

1

4

4

Real and equal roots: -2.0, -2.0

Enter coefficients

1

5

15

Imaginary roots are: $-2.5 + i2.9580$
and $-2.5 - i2.9580$

Enter coefficients

0

1

2

Equation is not quadratic

Q2. Develop a Java program to create a class Student with members usn, name, and array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;
class student{
    String USN , name;
    Scanner S1= new Scanner(System.in);
    int size = S1.nextInt();
    float credits[] = new float[size];
    float marks[] = new float[size];
    void accept(){
        USN= S1.next();
        name= S1.next();
        System.out.println("Marks in the following subjects are 1.Maths 2.physics
3.C prog 4.web 5.kannada 6.IDT 7.civil 8.english");
        for(int i=0;i<size;i++){
            System.out.print((i+1)+" = ");
            marks[i]= S1.nextInt();
        }
        System.out.println("respective credits of subjects are 1.Maths 2.physics
3.C prog 4.web 5.kannada 6.IDT 7.civil 8.english");
        for(int i=0;i<size;i++){
            System.out.print((i+1)+" = ");
            credits[i]= S1.nextInt();
        }
    }
    void display(){
        System.out.println("USN: "+USN+" name: "+name);
        System.out.println("Marks and credits in the following subjects are
1.Maths 2.physics 3.C prog 4.web 5.kannada 6.IDT 7.civil 8.english");
        for(int i=0;i<size;i++){
            System.out.print((i+1)+" .marks = "+marks[i]+" credits="+credits[i]+"
");
        }
    }
    int gpa(int i){
        if(marks[i]>=90) return 10;
        else if(marks[i]>=80 && marks[i]<=89) return 9;
        else if(marks[i]>=70 && marks[i]<=79) return 8;
        else if(marks[i]>=60 && marks[i]<=69) return 7;
```

```

        else if(marks[i]>=50 && marks[i]<=59) return 6;
        else if(marks[i]>=40 && marks[i]<=49) return 5;
        else if(marks[i]>=80 && marks[i]<=89) return 4;
        else return 0;
    }

    float sgpa(){
        float SGPA , sum=0;
        for(int i=0;i<size;i++){
            sum=sum+gpa(i)*credits[i];
        }
        //total credits=20
        SGPA = sum/20;
        return SGPA;
    }
}

class call{
    public static void main (String[] args) {
        student S1 = new student();
        S1.accept();
        S1.display();
        float Ans;
        Ans = S1.sgpa();
        System.out.println("SGPA of the student is : "+Ans);
    }
}

```


LAB-3

```
2) Import java.util.Scanner
class Book {
    String uost, name;
    Scanner s1 = new Scanner(System.in);
    int size = s1.nextInt();
    float credit[] = new float[size];
    float marks[] = new float[size];
    void accept() {
        uost = s1.next();
        name = s1.next();
        System.out.println("marks in the following
        subject 1. Maths 2. Physics 3. C Program
        4. English");
        for (int i = 0; i < size; i++) {
            System.out.print((i+1) + " = ");
            marks[i] = s1.nextFloat();
        }
        System.out.println("respective
        of credit of subject
        1. Maths 2. Physics 3. C Program
        4. Web 5. English");
    }
}
```

3

```
3.
int gpa(int i) {
    if (marks[i] >= 90) return 10;
    else if (marks[i] >= 70) {
        if (marks[i] >= 50) {
            if (marks[i] >= 40 && marks[i] <= 50) return 6;
            else return 5;
        }
    }
    else return 0;
}
```

```

float sgpa() {
    float sepa; sum = 0;
    for (int i = 0; i < size; i++) {
        sum = sum + gpa[i] + credits[i];
    }
    // total credits = 20
    sepa = sum / 20;
    return sepa; }

```

```

class class {
    psvm (String[] args)
    {
        student s1 = new student ();
        s1.accept ();
        s1.display ();
        float ans = s1.sgpa ();
        System.out.println ("sepa of the student")
    }
}

```

output -

Number of books: 3
 Enter details of Book:
 Name: abcd
 Author: ebn
 Price: 123
 Number of pages: 456

Q3. 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.

```
import java.util.Scanner;
class Book{
    String name, Author;
    int price,num_pages;
    Book(){};
    Book( String name,String Author,int price, int num_pages){
        this.name=name;
        this.Author=Author;
        this.price=price;
        this.num_pages=num_pages;
    }
    void set(){
        System.out.println("enter details of book :");
        Scanner S1= new Scanner(System.in);
        name = S1.next();
        Author = S1.next();
        price = S1.nextInt();
        num_pages= S1.nextInt();
    }
    void get(){
        System.out.println("name: "+name+" Author: "+Author+" price: "+price+"
num_pages: "+num_pages);
    }
    public String toString(){
        return("name: "+name+" Author: "+Author+" price: "+price+" num_pages:
"+num_pages);
    }
}
class bookdemo{
    public static void main (String[] args) {
        int n;
        Scanner S= new Scanner(System.in);
        System.out.print("print n: ");
        n=S.nextInt();
```



```

Book B[] = new Book[n];
B[0] = new Book("cant_hurt_me", "David_googins", 700, 360);
B[0].get();
for(int i=1; i<n; i++){
    B[i] = new Book();
    B[i].set();
}
System.out.println(B[1].toString());
for(int i=2; i<n; i++){
    B[i].get();
}
}

```

Date 22/11/2
Page 17

```

Scanner S = new Scanner(System.in);
System.out.println("Enter n:");
n = S.nextInt();
Book B[] = new Book[n];
B[0] = new Book("cant_hurt_me", "David_googins", 700, 360);
System.out.println(B[0].toString());
for(int i=1; i<n; i++){
    B[i] = new Book();
    B[i].set();
    B[i].get();
}
}

```

Enter name:
ABC

Marks in the following subjects are
1. physics 2. Python 3. Kannada 4. Jd T
7. electronics

1. 96 2. 99 3. 77 4. 84 5. 67 6. 89

Respectively credits of subjects
are 1. Maths 2. physics 3. Kannada
4. electronics 5. Jd T

1. 4 2. 4 3. 4 4. 5 5. 1

UOA: 127 Name ABC

Marks and credits in the following
1. Maths 2. Physics 3. Python 4. C 5.
Kannada 6. Jd T

| | |
|---------------|------------|
| 1. marks = 96 | credit = 4 |
| 2. marks = 97 | credit = 4 |
| 3. marks = 77 | credit = 3 |

Date / /
Page

22/1/21

```

Scanner S = new Scanner(System.in);
System.out.println("Enter n:");
n = S.nextInt();
Book B[] = new Book[n];
B[0] = new Book("C++", 100);
System.out.println(B[0].toString());
for (int i = 1; i < n; i++)
{
    B[i] = new Book();
    B[i] = set();
    B[i] = get();
}
    
```

Enter name:
ABC

Marks in the following subjects are

1. physics 2. Python 3. Kannada 4. Jdt
5. electronics

1. 96 2. 99 3. 77 4. 84 5. 67 6. 89

Respectively credits of subjects
are 1. Maths 2. physics 3. Kannada
4. electronics 5. Jdt

1. 4 2. 4 3. 4 4. 5 5. 1

Roll: 127 Name ABC

Marks and credits in the following

1. Maths 2. Physics 3. Python 4. C. 5.
Kannada 6. Jdt.

1. marks = 96
2. marks = 97
3. marks = 77

credit = 4
credit = 4
credit = 3

Q4. 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.

```
abstract class shape{
    int a,b;
    abstract void printarea();
}
class rectangle extends shape{
    rectangle(int x,int y){
        a=x;b=y;
    }
    void printarea(){
        System.out.println("area of rectangle is : "(a*b));
    }
}
class triangle extends shape{
    triangle(int x,int y){
        a=x;b=y;
    }
    void printarea(){
        System.out.println("area of triangle is : "(0.5*a*b));
    }
}
class circle extends shape{
    circle(int x){
        a=x;
    }
    void printarea(){
        System.out.println("area of circle is : "(3.14*a*a));
    }
}
class shapedemo{
    public static void main(String xx[]){
        rectangle r=new rectangle(5,4);
        triangle t=new triangle(5,4);
        circle c=new circle(7);
        r.printarea();
```



```

t.printarea();
c.printarea();
}
}

```

Q4 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.

→ abstract class shape

```

{
    int a1=10, a2=15;
    void printarea() {}
}

```

class rectangle extends shape

```

{
    void printarea()
    { System.out.println("The area of rectangle is : " + (a1 * a2)); }
}

```

class triangle extends shape

```

{
    void printarea()
    { System.out.println("The area of triangle is : " + (a1 * a2 * 0.5)); }
}

```

class circle extends shape

```

{
    void printarea()
    { System.out.println("The area of circle is : " + (3.14 * a1 * a2)); }
}

```


abstract
two
printArea()
e, Triangle
the challenge
the
area()
ape.

```
class abc  
{ public static void main (String xx[])  
{  
    rectangle s1 = new rectangle ();  
    triangle s2 = new triangle ();  
    circle s3 = new circle ();  
    s1.printArea();  
    s2.printArea();  
    s3.printArea();  
}}}
```

Output:

The area of rectangle is : 150

The area of triangle is : 75.0

The area of circle is : 814.0

81124

s6 :

s :

Q5. 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.

```
import java.util.Scanner;

class Account {
    String customerName;
    int accountNumber;
    String accountType;
    double balance;

    Account(String name, int accNo, String accType, double initialBalance) {
        customerName = name;
        accountNumber = accNo;
        accountType = accType;
        balance = initialBalance;
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit of $" + amount + " successful.");
    }

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

```

    }
}

class CurAcct extends Account {
    double minBalance;
    double serviceCharge;

    CurAcct(String name, int accNo, String accType, double initialBalance, double
minBal, double charge) {
        super(name, accNo, accType, initialBalance);
        minBalance = minBal;
        serviceCharge = charge;
    }

    void withdraw(double amount) {
        if (balance - amount >= minBalance) {
            balance -= amount;
            System.out.println("Withdrawal of $" + amount + " successful.");
        } else {
            System.out.println("Insufficient funds. Withdrawal failed.");
        }
    }

    void deductServiceCharge() {
        if (balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("Service charge of $" + serviceCharge + " applied
due to balance below minimum.");
        }
    }
}

class SavAcct extends Account {
    double interestRate;

    SavAcct(String name, int accNo, String accType, double initialBalance, double
interest) {
        super(name, accNo, accType, initialBalance);
        interestRate = interest;
    }

    void calculateInterest() {
        double interest = balance * interestRate / 100;
        balance += interest;
        System.out.println("Interest of $" + interest + " added.");
    }
}

```

```

    }

    void withdraw(double amount) {
        if (balance - amount >= 0) {
            balance -= amount;
            System.out.println("Withdrawal of $" + amount + " successful.");
        } else {
            System.out.println("Insufficient funds. Withdrawal failed.");
        }
    }
}

class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Creating savings account
        SavAcct savings = new SavAcct("John Doe", 123456, "Savings", 1000, 5); //
5% interest rate

        // Creating current account
        CurAcct current = new CurAcct("Jane Doe", 654321, "Current", 2000, 500,
10); // $500 minimum balance, $10 service charge

        System.out.println("Welcome to our bank!");

        while (true) {
            System.out.println("\n1. Deposit\n2. Withdraw\n3. Display Balance\n4.
Exit");

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

            switch (choice) {
                case 1:
                    System.out.print("Enter amount to deposit: ");
                    double depositAmount = scanner.nextDouble();
                    System.out.print("Select account (1 for Savings, 2 for
Current): ");

                    int accountChoice = scanner.nextInt();
                    if (accountChoice == 1)
                        savings.deposit(depositAmount);
                    else if (accountChoice == 2)
                        current.deposit(depositAmount);
                    break;
                case 2:

```


12/2/24

LAB-7

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

```
class Account {  
    String customerName;  
    int accountNumber;  
    String accountType;  
    double balance;
```

```
    Account(String name, int accNo,  
            String accType, double  
            initialBalance) {
```

```
        {  
            customerName = name;  
            accountNumber = accNo;  
            accountType = accType;  
            balance = initialBalance;  
        }  
    }
```

```
    void deposit(double amount) {
```

```
        balance += amount;  
        System.out.println("Deposit of $" +  
            amount + " successful");  
    }
```

```
    void displayBalance() {
```

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

```
}
```

```
class curAcct extends Account {  
    {
```

```
        double minBalance;  
        double serviceCharges;
```

Date: / /
Page:
curAcc + (String name, int accNo,
String accType, double initialBalance,
double minBal, double charge)

```
{  
    super(name, accNo, accType,  
        initialBalance);  
    minBalance = minBal;  
    serviceCharge = charge;  
}
```

void withdraw(double amount)

```
{  
    if (balance - amount >= minBalance)  
    {  
        balance -= amount;  
        System.out.println("Withdrawal  
of $ " + amount + " successful");  
    }  
}
```

else

```
{  
    System.out.println("Insufficient  
funds - withdrawal failed");  
}
```

}

void deductServiceCharge()

```
{  
    if (balance < minBalance)  
    {  
        balance -= serviceCharge;  
        System.out.println("Service charge  
of $ " + serviceCharge + " applied due to  
balance below minimum.");  
    }  
}
```

} }

Date: / /
Page:
class sav Acct extends Account {
double interestRate;

sav Acct (String name, int accNo,
String accType, double
initialBalance, double interest)
{

Super(name, accNo, accType,
initialBalance);
interestRate = interest;

24

void calculateInterest()

{

double interest = Balance *
interestRate / 100;

balance += interest;

System.out.println("Interest of \$"
+ interest + " added");

}

void withdraw (double amount)

{

if (balance - amount >= 0)

{

balance -= amount;

System.out.println("Withdrawal
of \$" + amount + " successful");

}

else

{

System.out.println("Insufficient
funds. Withdrawal failed");

}

}

}


```

class Bank {
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Welcome to our bank!");

        while (true)
        {
            System.out.println("1. Deposit 2. Withdraw 3. Display  
4. Balance 5. Exit");
            System.out.println("Enter your choice:");
            int choice = scanner.nextInt();
            switch (choice)
            {
                case 1: System.out.println("Enter the  
amount to deposit:");
                    double depositAmount = scanner.nextDouble();
                    System.out.println("Select account 1. for  
Savings, 2. for Current  
int account choice =  
scanner.nextInt();
                    scanner.nextInt();
                    if (account choice == 1)
                        Savings.deposit(depositAmount);
                    else if (account choice == 2)
                        Current.deposit(depositAmount);
                    break;
                case 2: System.out.println("Enter the  
amount to be withdrawn");
                    double withdrawnAmount =
                        scanner.nextDouble();

```

Date: / /
Page:

```

system.out.println("select account
(1. for saving 2. for current);");
int account choice = scanner.nextInt();
if (account choice
scanner.nextInt();
    if (account choice == 1)
        savings.withdrawal(withdrawnAmount);
    else if (account choice == 2)
        current.withdrawal(withdrawn
            Amount);
        current.deductServiceCharge();
        break;

```

```

case 3: system.out.println("select
account (1 for savings 2. for current)
account choice =
scanner.nextInt();
    if (account choice == 1)
        savings.displayBalance();
    else if (account choice == 2)
        current.displayBalance();
        break;

```

```

case 4: system.out.println("Thank you
for Banking with us!");
system.exit(0);
default:

```

```

system.out.println("Invalid choice.
please try again");

```

}

}

}

}

Date: / /
Page:
- output -

Welcome to our bank !

1. Deposit
2. Withdrawal
3. Display Balance
4. Exit

Enter your choice: 1

Enter amount to deposit: 2000

Select account (1. Savings, 2. Current) 1

Enter your choice: 1

Enter amount to deposit: 1500

Select account (1. Savings, 2. Current) 2

Enter choice: 3

Select account: 1

Balance: Rs 3000.00

Enter choice: 3

Select account: 2

Balance: Rs 3500.00

Enter choice: 2

Select account: 1 Enter amount: 2000

Withdrawal of Rs 6000.00 - overpaid

Enter choice: 4

Q6. Create a package CIE which has two classes- Student and Internals. The class Student 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.

```
package CIE;
import java.util.Scanner;
public class student
{
    public String USN,name;
    public int semester;
    Scanner S1 = new Scanner(System.in);
    public void set()
    {
        System.out.println("USN of the student is : ");
        USN = S1.next();
        System.out.println("name of the student is : ");
        name = S1.next();
        System.out.println("semester of the student is : ");
        semester = S1.nextInt();
    }
    public void get()
    {
        System.out.println("USN of the student is : "+USN+"name of the student is : "+name+"semester of the student is : "+semester);
    }
}

package CIE;
import java.util.Scanner;
public class internal extends student
{
    Scanner S1 = new Scanner(System.in);
```



```

public int internal_marks[]= new int[5];
public void setcie()
{
    System.out.println("1.Java 2.maths 3.DS 4.COA 5.DBMS ");
    for(int i=0;i<5;i++)
    {
        System.out.print("\n"+(i+1)+".");
        internal_marks[i]= S1.nextInt();
    }
}
public void getcie()
{
    System.out.println("1.Java 2.maths 3.DS 4.COA 5.DBMS ");
    for(int i=0;i<5;i++)
    {
        System.out.println((i+1)+". "+internal_marks[i]);
    }
}
}
package SEE;
import CIE.student;
import java.util.Scanner;
public class external extends CIE.student
{
    Scanner s1 = new Scanner(System.in);
    public int see_marks[]=new int[5];
    public void setsee()
    {
        System.out.println("1.Java 2.maths 3.DS 4.COA 5.DBMS ");
        for(int i=0;i<5;i++)
        {
            System.out.print("\n"+(i+1)+".");
            see_marks[i]= s1.nextInt();
        }
    }
    public void getsee()
    {
        System.out.println(" 1.Java 2.maths 3.DS 4.COA 5.DBMS ");
        for(int i=0;i<5;i++)
        {
            System.out.println(i+ ". "+see_marks[i]);
        }
    }
}

```

```

}

import java.util.Scanner;
import CIE.student;
import CIE.internal;
import SEE.external;
public class fmarks
{
    public static void main(String XX[] )
    {
        int n;
        System.out.println("enter number of students : ");
        Scanner S1=new Scanner(System.in);
        n=S1.nextInt();
        CIE.student S[]=new student[n];
        CIE.internal I[]= new internal[n];
        SEE.external E[]= new external[n];
        for(int i=0;i<n;i++)
        {
            S[i] = new student();
            I[i]= new internal();
            E[i]=new external();
            System.out.println("Enter details of student : ");
            S[i].set();
            System.out.println("Enter internal marks of student (out of 50) : ");
            I[i].setcie();
            System.out.println("Enter see marks of student (out of 100) : ");
            E[i].setsee();
        }
        for(int i=0;i<n;i++)
        {
            System.out.println("details of student are : ");
            S[i].get();
            System.out.println("internal marks of student are : ");
            I[i].getcic();
            System.out.println("see marks of student are : ");
            E[i].getsee();
        }
        //final marks calculation
        for (int i = 0; i < n; i++) {
            System.out.println("Marks of student " + S[i].name + ":");
            for (int j = 0; j < 5; j++) {
                int subjectTotalMarks = I[i].internal_marks[j] +
                ((E[i].see_marks[j])/2);
            }
        }
    }
}

```

```

        System.out.println("Subject " + (j + 1) + " marks: " +
subjectTotalMarks);
    }
    System.out.println();
}
}
}

```

29/4/24

LAB-5

Date ___/___/___
Page ___

2) Package CTF;

import java.util.Scanner;

public class Student {

public String usn;

public String name;

public int sem;

public void accept() {

Scanner s = new Scanner(System.in);

System.out.println("Enter Name");

this.name = s.nextLine();

System.out.println("Enter usn");

this.usn = s.nextLine();

System.out.println("Enter sem");

this.sem = s.nextInt();

}

public void display() {

System.out.println("Name: " + this.name +

" usn: " + this.usn + " sem: " +

this.sem);

}

}

Package CTF;

import java.util.Scanner;

public class internal extends CTF Student {

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

CTF Student student = new

CTF Student();

public void accept() {

student.accept();

```
Scanner s1 = new  
Scanner(System.in);  
System.out.println("Enter Internal  
Marks:");  
for (int i = 0; i < 5; i++)  
{  
    m[i] = s1.nextInt();  
}
```

```
}  
  
Public void display ()  
{  
    student.display();  
    for (int i = 0; i < 5; i++)  
    {  
        System.out.println("Marks of sub" +  
            (i + 1) + " = " + m[i]);  
    }  
}
```

```
}
```

```
}
```

output -

Enter number of students: 2

Enter student details:

ABDE

CS115

3

Enter ~~IE~~ marks:

40

39

38

35

30

Enter the SEE marks

98

97

93

92

91

85

Enter student details

EFE, HI

CS 128

3

Enter CIE MARKS

25 26 29 32 35

Enter SEE Marks

87 97 94 92 76

Student Details :

Name : ABCDE

USN : C6127 / 1111 111111

Semester : 4

Marks obtained in CIE Subject 1 : 90

Marks obtained in CIE Subject 2 : 92

Marks obtained in CIE Subject 3 : 87

Marks obtained in CIE Subject 4 : 84

Marks obtained in CIE Subject 5 : 97

Marks obtained in CIE Subject 6 : 96

Q7. 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=father’s age.

```
class wrongageexception extends Exception{
    wrongageexception(String message){
        super(message);
    }
    public String toString(){
        return "wrong age enetered";
    }
}
class father{
    int age;
    father(int age) throws wrongageexception{
        if(age<0){
            throw new wrongageexception("age cannot be negative");
        }
        this.age=age;
    }
}
class son extends father{
    int sonage;
    son(int fatherage, int sonage) throws wrongageexception{
        super(fatherage);
        if(sonage >= fatherage){
            throw new wrongageexception("son age cannot be greater than father
age");
        }
        this.sonage=sonage;
    }
}
class exceptiondemo{
    public static void main(String xx[])
    {
        try{
            father f=new father(40);
            son s=new son(f.age,25);
        }
    }
}
```

```

        System.out.println("father age :"+f.age);
        System.out.println("son age :"+s.sonage);
    }
    catch(wrongageexception e){
        System.out.println("exception:"+e.toString());
        System.out.println("exception:"+e.getMessage());
    }
}
}
}

```

LNB-4

Date: / /
Page: /

22/11/24

```

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

class Father{
    private int age;

    public Father (int age) throws WrongAgeException
    {
        if (age < 0)
        {
            throw new WrongAgeException ("Age cannot be negative");
        }
        this.age = age;
    }

    public int getAge(){
        return age;
    }
}

class Son extends Father{
    private int sonAge;

    public Son (int FatherAge, int sonAge) throws WrongAgeException{
        super (FatherAge);

        if (sonAge >= FatherAge)
            throw new WrongAgeException ("son's age should be less than Father's age.");
    }
}

```



```

    }
    this.SonAge = sonAge;
    }
    public int getSonAge() {
        return sonAge;
    }
    }

```

```

    }
    class Main {
        public static void main (String[] args) {
            try {
                Father father = new Father (45);
                Son son = new Son (father.getAge(), 20);

                System.out.println ("Father's age: " + father.getAge());
                System.out.println ("Son's age: " + son.getSonAge());
            }
            catch (WrongAgeException e) {
                System.out.println ("Exception: " + e.getMessage());
            }
        }
    }
}

```

- output -

Father's age: 45
Son's age: 20

Q8. Write a program to create two threads. One thread displays "BMS college of Engineering" once every ten seconds and another displays "CSE" once every two seconds.

```
class BMSthread implements Runnable{
    public void run(){
        while(true){
            try{
                System.out.println("BMS College of engineering");
                Thread.sleep(10000);
            }
            catch(InterruptedException ie){
                System.out.println("Thread Interrupted");
            }
        }
    }
}

class CSEthread implements Runnable{
    public void run(){
        while(true){
            try{
                System.out.println("CSE");
                Thread.sleep(2000);
            }
            catch(InterruptedException ie){
                System.out.println("Thread Interrupted");
            }
        }
    }
}

class display{
    public static void main(String xx[]){
        thread bms=new thread(new BMSthread);
        thread cse=new thread(new CSEthread);
        bms.start();
        cse.start();
    }
}
```

5/2/24

LAB-6

Date ____/____/____
Page ____

- * Write a program to create 2 threads one thread displaying "BMS College of Engineering" once every second and another displaying "CSE" once every two seconds.

```
class BMS Thread implements Runnable
{
    public void run()
    {
        while (true)
        {
            try
            {
                System.out.println("BMS College of Eng");
                Thread.sleep(1000);
            }
            catch (InterruptedException e)
            {
                System.out.println("BMS Thread is interrupted");
            }
        }
    }
}
```

```
class CSE Thread implements Runnable
{
    public void run()
    {
        while (true)
```

Date: / /
Page:

```

    {
        try
        {
            System.out.println("CE");
            Thread.sleep(2000);
        }
        catch (InterruptedException e) {}
    }
}

Public class Display
{
    Public static void main (String[] args)
    {
        Thread bms = new Thread
            (new BMSThread());
        Thread cse = new Thread
            (new CSEThread());
        bms.start();
        cse.start();
    }
}

```

OUTPUT -

BMS college of Engineering
CSE

CSE

CSE

CSE

Q9. Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a `NumberFormatException`. If Num2 were Zero, the program would throw an `ArithmeticException`. Display the exception in a message dialog box.

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

public class ExceptionDemo extends JFrame implements ActionListener {
    private JTextField t1, t2, t3;
    private JLabel l1, l2;
    private JButton b1, b2;

    public ExceptionDemo() {
        setLayout(new FlowLayout());
        l1 = new JLabel("Num1 :");
        add(l1);
        t1 = new JTextField(5);
        add(t1);
        l2 = new JLabel("Num2 :");
        add(l2);
        t2 = new JTextField(5);
        add(t2);
        t3 = new JTextField(5);
        t3.setEditable(false);
        add(t3);
        b1 = new JButton("Divide");
        add(b1);
        b1.addActionListener(this);
        b2 = new JButton("Clear");
        add(b2);
        b2.addActionListener(this);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 150); // Set an appropriate size
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        // Implementation of the divide logic and exception handling
    }
}
```



```
setVisible(true); }  
public void actionPerformed(ActionEvent act) {  
    String str = act.getActionCommand();  
    if (str.equals("Divide")) {  
        try {  
            int num1 = Integer.parseInt(t1.getText());  
            int num2 = Integer.parseInt(t2.getText());  
            int num3 = num1 / num2;  
            t3.setText("" + num3);  
        } catch (ArithmeticException e) {  
            JOptionPane.showMessageDialog(this, "ArithmeticException: Cannot divide by  
zero!");  
        } catch (NumberFormatException e) {  
            JOptionPane.showMessageDialog(this, "NumberFormatException: Please enter valid  
integers for Num1 and Num2." }  
        } else {  
            t1.setText("");  
            t2.setText("");  
            t3.setText(""); } }  
public static void main(String[] args) {  
    SwingUtilities.invokeLater(new Runnable() {  
        public void run() {  
            new ExceptionDemo();  
        } }));  
}
```

Report For Applet Programs:

19/2/24

LAB-8

Date / /
Page

* AWT Programs - RECAP

- (1) Button drag.java: It opens a Button game window with 3x3 tiles with numbers and provides user with buttons for reset, start and restart. Once clicked on start, user can click on two tiles and simultaneously clicked tiles to get swapped. User can click on reset to start the game again.
- (2) ButtonList.java: It opens a Button list window with three buttons - yes, no, undecided and has a default text Hello. Once clicking on yes, the window moves on the text - You pressed yes - Similarly on clicking No -> Pressed No and on clicking undecided -> You Pressed undecided.
- (3) ButtonList.java: It opens Button window with 3 buttons - yes, no, undecided. On clicking yes, a Dialog window appears with message you pressed yes and on button. Similarly on clicking no, you pressed No and on clicking undecided, you pressed undecided in separate dialog box.

④ Division Main.java - It opens Division of Integers window with fields to enter 2 numbers and a button. Result on clicking which the two numbers quotient will appear on the button after result. It gives and takes numbers in float data type.

⑤ Division Main1.java - It opens a Division of integers window similar to last program. But the result provided will be integer numbers & and quotient in float.

⑥ Text Field Demo.java - It opens a TF-Label Demo window with fields for entering name and password on clicking enter after entering the name, the text box name appear after Name. Similarly on clicking enter after entering the password, it appear text to password. on selecting some letters in name field and clicking enter, the selected text appears next to select text in name. Password reverses. The input and display it after encrypting it.

21/2/24