

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 Main
{
    public static void main(String xx[]){
        float a,b,c,d,r1,r2;
        Scanner sl=new Scanner(System.in);
        System.out.println("enter the coefficients");
        a=sl.nextFloat();
        b=sl.nextFloat();
        c=sl.nextFloat();
        d=b*b-(4*a*c);
        if(d>0){
            r1=(-b+(float)Math.sqrt(d))/(2*a);
            r2=(-b-(float)Math.sqrt(d))/(2*a);
            System.out.println("r1="+r1);
            System.out.println("r2="+r2);
        }
        else if(d==0){
            r1=(-b)/(2*a);
            r2=(-b)/(2*a);
            System.out.println("r1="+r1);
            System.out.println("r2="+r2);
        }
        else{
            r1=(float)Math.sqrt(-d)/(2*a);
            r2=-1*r1;
            System.out.println("roots:\n"+ "r1=" + (-b/(2*a)) + "+" + r1 + "i" + "\nr2=" + (-b/(2*a)) + r2 + "i");
        }
        sl.close();
    }
}
```

```
enter the coefficients
1
2
3
roots:
r1=-1.0+1.4142135i
r2=-1.0-1.4142135i
```

## Lab - 1

Page No. \_\_\_\_\_  
Date \_\_\_\_\_

```
import java.util.Scanner;           question " "
import java.lang.Math;             & date
```

## public class QuadEqn {

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

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

`float a = s1.nextFloat();`

float b := 81. nextfloat();

float c = Sl.nextFloat();

float x1, y1; // position

```
'float d= (float) Math.sqrt(b*b - 4*a*c));  
if(d>0){
```

$$x_1 = (-b + \sqrt{b^2 - 4ac}) / (2*a);$$

```
 $\lambda_2 = (-b - \text{float}(\text{Math.sqrt}(cd))) / (2 * a);$ 
```

```
System.out.println("x1=" + x1 + " x2=" + x2),
```

else if (d == 0) {

$$x_1 = -b/(2*a)$$

825 = 81000 plus 1500

```
System.out.println("λ1=" + λ1 + "λ2=" + λ2)
```

3

? else if (num > max) max = num;

$r1 = \text{float}(\text{Math.sqrt}(-d)/(2*a))$

$$x_2 = -1 \star x_1 ;$$

1

五

~~18/2/22~~ 33

## Output

~~Chlorophytaceae~~

~~Chromatography~~

~~1. Cis-1,4-pentadiene~~

-5 1 2

~~6~~ ~~What's it like? 2 well 31.~~

$$(-\infty, -2) \cup (-1, \infty)$$

$$\therefore z = 0.5 + 0.285i$$

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.

```
class PRACT_CLLG {
    String usn,name;
    int [] credits = new int[8];
    float [] marks = new float[8];
    public void accept_name_usn(String name,String usn) {
        this.usn=usn;
        this.name=name;
    }
    public void accept_credits(int [] credits) {
        this.credits=credits;
    }
    public void accept_marks(float[] marks) {
        this.marks=marks;
    }
    public String return_usn() {
        return this.usn;
    }
    public String return_name() {
        return this.name;
    }
    public int [] return_credits() {
        return this.credits;
    }
    public float [] return_marks() {
        return this.marks;
    }
    public float sgpa(int[] credits, float[] marks) {
        int total_credits=0;
        float marks_credits=0;
        for(int i=0; i<8; i++) {
            total_credits=total_credits+credits[i];
        }
        for(int i=0; i<8; i++) {
            marks_credits=marks_credits+(credits[i]*marks[i]);
        }
        return marks_credits/total_credits;
    }
}
```

```
public static void main(String[] args) {
    PRACT_CLLG p = new PRACT_CLLG();
    p.accept_name_usn("Mukund", "lbtm22cs166");
    int [] credits_3rd={4,4,4,3,3,2,1,1};
    p.accept_credits(credits_3rd);
    float [] marks_3rd = {23.45678f, 45.789012f, 1.2345678f, 87.65432f, 9.012345f, 65.4321f, 34.56789f, 78.90123f};
    p.accept_marks(marks_3rd);
    System.out.println("usn: "+ p.return_usn());
    System.out.println("name: "+ p.return_name());
    int[] get_credits=p.return_credits();
    for(int i=0; i<8; i++) {
        System.out.printf("credits:%d ", get_credits[i]);
    }
    System.out.println();
    float[] get_marks=p.return_marks();
    for(int i=0; i<8; i++) {
        System.out.printf("marks:%f ", get_marks[i]);
    }
    System.out.println();
    System.out.println("sgpa: "+ p.sgpa(get_credits,get_marks));
}
```

{}

```
usn: lbm22cs166
nmae: Mukund
credits:4 credits:4 credits:4 credits:3 credits:3 credits:2 credits:1 credits:1
marks:23.456779 marks:45.789013 marks:1.234568 marks:87.654320 marks:9.012345 marks:65.432098 marks:34.567890 marks:78.901230
sgpa: 37.10249
```

## Lab - 2

Page No. \_\_\_\_\_  
Date: \_\_\_\_\_

- ① A java program to create class student usn, name, marks, credits and display details and sgpa

Ans public class Sgpa

    String usn, name;

    int [ ] credits = new int [8];

    float [ ] marks = new float [8];

    public void accept\_name\_usn (String name,  
        this.usn = usn ;

        this.name = name ;

}

    public void accept\_marks (float [ ] marks)  
        this.marks = marks ;

}

    public void accept\_credits (int [ ] credits)  
        this.credits = credits ;

}

    public String return\_usn () {  
        return this.usn ;

}

    public String return\_name () {  
        return this.name ;

}

    public int [ ] return\_credits () {  
        return this.credits ;

}

    public float [ ] return\_marks () {  
        return this.marks ;

}

    public float sgpa (int [ ] credits, float [ ]  
        int total\_credit = 0 ;  
        float marks\_credit = 0 ;

```

for (int i=0; i<8; i++) {
    total_credit = total_credit + credits[i];
}

for (int i=0; i<8; i++) {
    marks_credits = marks_credits +
        (credits[i] * marks[i]);
}

return marks_credits / total_credit;
}

public static void main (String [] args) {
    SGPA p = new SGPA();
    p.accept_name("Mukund", "1bm22cs166");
    int [] credits_3rd = {4, 4, 4, 3, 3, 2, 1, 1};
    p.accept_credits(credits_3rd);
    float [] marks_3rd = {23.4f, 45.78f,
        1.23f, 87.6f, 9.01f, 65.43f,
        34.56f, 78.3f};

    p.accept_marks(marks_3rd);
    System.out.println("UIN: " + p.return_uin);
    System.out.println("Name: " + p.return_name);
    int [] get_credits = p.return_credits;
    for (int i=0; i<8; i++) {
        System.out.print("marks: " + get_marks(i));
    }
    System.out.println();
    p.SGPA(get_credits, get_marks);
}

```

marks % Output

name: Mukund  
uin: 1bm22cs166

66 \_\_\_\_\_  
Page No. \_\_\_\_\_  
Date: 1 1 , „

marks: 23.45, 14.45, 45

sgpa: 3.71

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(String n,String a,int p,int np){
        this.name=n;this.author=a;
        this.price=p;this.num_pages=np;
    }
    public String toString(){
        return "Book name:"+name+"\nAuthor:"+author+"\nPrice:"+price+"\nNumber of pages:"+num_pages;
    }
}
class BookDet {
    static Scanner s=new Scanner(System.in);
    static Book set(){
        s.nextLine();
        System.out.println("Enter book name");
        String n=s.nextLine();
        System.out.println("Enter author name");
        String a=s.nextLine();
        System.out.println("Enter price of book");
        int p=s.nextInt();
        System.out.println("Enter no of pages");
        int np=s.nextInt();
        Book bl=new Book(n,a,p,np);
        return bl;
    }
    public static void main(String sx[]){
        int n;
        System.out.println("Enter no of books");
        n=s.nextInt();
        Book b[]=new Book[n];
        for(int i=0;i<n;i++)
            b[i]=set();
        System.out.println("Details of books entered");
        for(int i=0;i<n;i++)
            System.out.println(b[i]);
    }
}
```

Enter no of books

1

Enter book name

berserk

Enter author name

kentaro miura

Enter price of book

1000

Enter no of pages

200

Details of books entered

Book name:berserk

Author:kentaro miura

Price:1000

Number of pages:200

Q) Create a book class and create a constructor, get and set methods and make a method to create an array of objects.

Soln

```
import java.util.Scanner;
class Book {
    String name, author;
    int numPages, price;
    Book(String n, String a, int np, int p) {
        this.name = n;
        this.author = a;
        this.numPages = np;
        this.price = p;
    }
    public String toString() {
        return "Name: " + name + " Author: " +
            author + " Price: " + price + " noPage: " +
            numPages;
    }
}
```

public class BookDet

```
static Scanner s = Scanner(System.in);
static Book set() {
    s.nextLine();
    System.out.println("Enter book");
    String n = s.nextLine();
```

```

8 System.out.println("author: ");
String a = s.nextLine();
System.out.println("Enter of book");
int p = s.nextInt();
if (p < 0) {
    System.out.println("Invalid price");
    System.exit(1);
}
    
```

3.

```

System.out.println("No. page: ");
int np = s.nextInt();
if (np < 0) {
    System.out.println("invalid ");
    System.exit(1);
}
    
```

3

```

Book b1 = new Book(n, a, np, p);
return b1;
}
    
```

3

```

public static void main(String[] args) {
    int n;
    System.out.println("Mukund, Business");
    System.out.println("enter no. of books");
    n = s.nextInt();
    Book b[] = new Book[n];
    for (int i = 0; i < n; i++) {
        b[i] = set();
        System.out.println("detail:");
        for (int j = 0; j < n; j++) {
            System.out.println(b[j]);
        }
    }
}
    
```

3

3.

Output

Enter no. of books

21

Enter book name

Berserk

Enter author name

Kentaro Miura

Enter price

1000

Enter number of pages

200

Enter book name

Name: Berserk

author: Kentaro Miura

price: 1000

Number of pages: 100

09/10

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 print Area( ) that prints the area of the given shape.

```
abstract class shape{
    int x=0,y=0;
    public shape(int x, int y) {
        this.x = x;
        this.y = y;
    }
    abstract void printarea() ;
}

class rect extends shape{
    public rect(int l, int b){
        super(l,b);
    }
    public void printarea(){
        System.out.println("area of rectangle: " + (x*y));
    }
}

class tri extends shape{
    public tri(int b, int h){
        super(b,h);
    }
    public void printarea(){
        System.out.println("area of triangle: " + (x*y*0.5));
    }
}

class cir extends shape{
    public cir(int r){
        super(r,r);
    }
    public void printarea(){
        System.out.println("area of circle: " + (3.14*x*x));
    }
}

class shape_abs{
    public static void main(String[] args){
        rect rl= new rect(20,3);
        tri tl = new tri(5,10);
        cir cl = new cir(4);
        rl.printarea();
        tl.printarea();
        cl.printarea();
    }
}
```

area of rectangle: 60

area of triangle: 25.0

area of circle: 50.24

## Lab-3

(1) Develop a java program to create abstract class shape that contains two integers and an empty method named printArea(). Provide three classes rect, tri, cir, such that each of them extends to shape. Each class prints the method printArea().

Soln abstract class shape {

```
int x=0, y=0;
public shape (int x, int y) {
    this.x = x;
    this.y = y;
}
```

abstract void printArea();

}

class rect extends shape {

```
public rect (int l, int b) {
    super (l, b);
}
```

public void printArea() {

```
System.out.print ("area : " +(x*y));
}
```

}

class tri extends shape {

```
public tri (int b, int h) {
    super (b, h);
}
```

public void printArea() {

```
System.out.println ("area : " +(x*y/2));
}
```

}

class cir extends shape

```
public cir (int r) {
```

```
super(r, r);
```

```
public void printarea() {
```

```
System.out.println("Area is " + (3.14 * r * r));
```

}

class shape-abs {

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

```
rect r1 = new rect (20, 3),
```

```
tri t1 = new tri (5, 10);
```

```
cir c1 = new cir (4);
```

```
t1.printarea();
```

```
t1.printarea();
```

```
c1.printarea();
```

}

Output

area of rectangle: 60

area of triangle: 25.0

area of circle: 50.24

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;
    long accountNumber;
    String accountType;
    double balance;

    public Account(String customerName, long accountNumber, String accountType) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = 0.0;
    }

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

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

    public void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient balance.");
        } else {
            balance -= amount;
            System.out.println("Withdrawal successful.");
        }
    }
}
```

```
class Sav_acct extends Account {
    double interestRate;

    public Sav_acct(String customerName, long accountNumber) {
        super(customerName, accountNumber, "Savings");
        this.interestRate = 0.05; // Assuming 5% interest rate
    }

    public void depositInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest deposited: " + interest);
    }
}

class Cur_acct extends Account {
    double minimumBalance;
    double serviceCharge;

    public Cur_acct(String customerName, long accountNumber) {
        super(customerName, accountNumber, "Current");
        this.minimumBalance = 1000.0; // Minimum balance requirement
        this.serviceCharge = 50.0; // Service charge for falling below minimum balance
    }

    @Override
    public void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient balance.");
        } else {
            balance -= amount;
            if (balance < minimumBalance) {
                balance -= serviceCharge;
                System.out.println("Service charge applied.");
            }
            System.out.println("Withdrawal successful.");
        }
    }
}
```

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

        // Creating savings account
        Sav_acct savingsAccount = new Sav_acct("John Doe", 123456789);

        // Creating current account
        Cur_acct currentAccount = new Cur_acct("Jane Doe", 987654321);

        // Deposit to savings account
        System.out.print("Enter amount to deposit to savings account: ");
        double depositAmount = scanner.nextDouble();
        savingsAccount.deposit(depositAmount);
        savingsAccount.displayBalance();

        // Deposit to current account
        System.out.print("Enter amount to deposit to current account: ");
        depositAmount = scanner.nextDouble();
        currentAccount.deposit(depositAmount);
        currentAccount.displayBalance();

        // Withdraw from savings account
        System.out.print("Enter amount to withdraw from savings account: ");
        double withdrawAmount = scanner.nextDouble();
        savingsAccount.withdraw(withdrawAmount);
        savingsAccount.displayBalance();

        // Withdraw from current account
        System.out.print("Enter amount to withdraw from current account: ");
        withdrawAmount = scanner.nextDouble();
        currentAccount.withdraw(withdrawAmount);
        currentAccount.displayBalance();

        // Deposit interest to savings account
        savingsAccount.depositInterest();
        savingsAccount.displayBalance();

        scanner.close();
    }
}
```

---

Enter amount to deposit to savings account: 20000

Deposit successful.

Current Balance: 20000.0

Enter amount to deposit to current account: 30000

Deposit successful.

Current Balance: 30000.0

Enter amount to withdraw from savings account: 40

Withdrawal successful.

Current Balance: 19960.0

Enter amount to withdraw from current account: 2000

Withdrawal successful.

Current Balance: 28000.0

Interest deposited: 998.0

Current Balance: 20958.0

## Lab-8

" \_\_\_\_\_  
Page No. \_\_\_\_\_  
Date: \_\_\_\_\_",

### Bank program

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

```
class Account {
```

```
String customerName;
```

```
long accountNumber;
```

```
String accountType;
```

```
double balance
```

```
public Account (String cn, long an, String at) {  
    this.customerName = cn;  
    this.accountNumber = an;  
    this.accountType = at;  
    this.balance = 0.0;
```

```
}
```

```
public void deposit (double amount) {  
    balance += amount;
```

```
    System.out.println ("Deposit successful");
```

```
public void displayBalance () {  
    System.out.println ("Current Balance : " + bal);
```

```
public void withdraw (double amount) {  
    if (amount > balance) {
```

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

```
    } else {
```

```
        balance -= amount;
```

```
        System.out.println ("withdrawn successfully");
```

class Sav-Acc extends Accounts &  
 double interestRate;

public Sav-Acc (String cn, long an);  
 super (cn, an, "Savings");  
 this.interestRate = 0.05;

3

public void depositInterest () &  
 double interest = balance \* interestRate;  
 balance += interest;  
 System.out.println ("Interest dep.

3

class Curr-Acc extends Account &  
 double minimumBalance;  
 double serviceCharge;

public Curr-Acc (String customerName,  
 super (cn, an, "Current");  
 this.minimumBalance = 1000.0;  
 this.serviceCharge = 50.0;

3

@Override

public void withdraw (double amount)  
 if (amount > balance) &  
 System.out.println ("Insufficient

3

else &

balance -= amount;

if (balance < minimumBalance)  
 balance -= serviceCharge

System.out.println ("Applied")

public class Bank {

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

        Sav-Acc Savings = new Sav-Acc(123),

        Cur-Acc Current = new Cur-Acc("TD", 999)

        System.out.println("Enter amt : ");  
        depositAmount = s.nextDouble();  
        currentAmount = deposit(depositAmount);  
        currentAccount.displayBalance();

        System.out.print("Enter amt : ");  
        withdrawAmount = s.nextDouble();  
        currentAccount.withdraw(withdrawAmount);  
        currentAccount.displayBalance();

(long ans)

        System.out.print("Enter ");  
        double withdrawAmount = s.nextDouble();  
        Savings Account.withdraw(withdrawAmount);  
        Current Account.displayBalance();

        System.out.print("Enter ");  
        withdrawAmount = scanner.nextDouble();  
        Current Account.withdraw(withdrawAmount);  
        Current Account.displayBalance();

3

3

## Output

enter amount to deposit to savings : 20000

deposit successful.

Current Balance : 200000

enter amount to deposit to current : 30000

deposit successful.

Current Balance : 300000

enter amount to withdraw from savings :

Current Balance = 19960

enter amount to withdraw from current :

Current Balance : 28000.0

Interest deposited : 998.0

Current Balance : 20958

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 checks both father and son's age and throws an exception if son's age is >=father's age.

```
class MyException extends Exception {
    int detail;
    public MyException(int age, String exe) {
        this.detail=age;
        System.out.println(exe + " given age is: " + age + " please enter again");
    }
    public String getMessage() {
        return "Exception: " + detail;
    }
}
class Father{
    int age;
    public Father(int age) throws MyException {
        if (age < 0) throw new MyException(age , "Age cannot be lesser than 0");
        this.age = age ;
    }
}
class Son extends Father{
    int age;
    public Son(int fatherAge, int sonAge) throws MyException{
        super(fatherAge);
        this.age = sonAge;
        if (this.age > super.age) throw new MyException(age , "Age of son cannot be more than father");
    }
}
class father_son {
    public static void main(String[] args) {
        try {
            Father fl = new Father(-1);
            Son sl = new Son(30 , 31);
        }
        catch (MyException e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
    }
}
```

Age cannot be lesser than 0 given age is: -1 please enter again  
Exception caught: Exception: -1

②

Write a program handling exceptions in a inheritance tree with base class Father and subclass Son. Implement Constructors and throw exceptions for wrong age and for son class if son's age is greater than father's.

Class MyException extends Exception

int detail;

```
public MyException(int age, String exc) {
    this.detail = age;
    System.out.println(exc + " " + age);
}
```

```
public String getMessage() {
    return "Exception" + detail;
}
```

}

class Father {

int age;

```
public Father(int age) throws MyException {
    if (age < 0)
        throw new MyException(age,
            this.age = age);
}
```

}

class Son extends Father {

int age;

```
public Son extends Father {
    super.gatherAge();
    this.age = sonAge;
    if (this.age > super.age)
        throw new MyException(age);
```

threw new MyException(age);

```
public class Father_Son {  
    public static void main(String[] args) {  
        try {  
            Father f = new Father(-1);  
            Son s1 = new Son(30, 31);  
            catch (MyException e) {  
                System.out.println("____");  
            }  
        }  
    }  
}
```

Age cannot be lesser than given age is: 0  
please enter again.

Exception caught: Exception: 0

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.

```
package CIE;

public class student {
    public String usn;
    public String name;
    public int sem;
    public student(String u, String n, int s) {
        this.usn = u;
        this.name = n;
        this.sem = s;
    }
}
```

```
package CIE;

public class internals extends CIE.student{
    public double imarks[] ;
    public internals(String u, String n, int s ,double m[]) {
        super(u,n,s);
        this.imarks = m;
    }
}
```

```
package SEE;
import CIE.student;
public class externals extends CIE.student{
    public double smarks[];
    public externals (String u, String n, int s ,double m[]) {
        super(u,n,s);
        this.smarks = m;
    }
}
```

```
package result;
import CIE.student ;□
public class test {
    public static void main(String[] args) {
        double internal[] = {43, 45, 47, 44, 41};
        double external[] = {90, 87, 65, 98, 43};
        student sl = new student("1BM22CS166", "Mukund", 3);
        internals il = new internals("1BM22CS166", "Mukund", 3, internal);
        externals el = new externals("1BM22CS166", "Mukund", 3, external);
        System.out.println("usn:" + sl.usn + " name: " + sl.name + " sem: " + sl.sem );
        System.out.println("internal marks: ");
        for(int i = 0; i<5 ; i++) {
            System.out.printf("internal marks %d : %f", i+1, il.imarks[i]);
        }
        System.out.println();
        System.out.println("external marks: ");
        for(int i = 0; i<5 ; i++) {
            System.out.printf("external marks %d : %f", i+1, el.smarks[i]);
        }
    }
}
```

```
usn:1BM22CS166 name: Mukund sem: 3
internal marks:
internal marks 1 : 43.000000internal marks 2 : 45.000000internal marks 3 : 47.000000internal marks 4 : 44.000000internal marks 5 : 41.000000
external marks:
external marks 1 : 90.000000external marks 2 : 87.000000external marks 3 : 65.000000external marks 4 : 98.000000external marks 5 : 43.000000
```

## Lab-5

Page No. \_\_\_\_\_  
Date: \_\_\_\_\_

- ① Create a package CIE which has student and internals. student has personal information and internals has an array of 5 marks containing marks. Create a package SEE which has final marks of 5 courses in an array. import two packages and declare

CIE/student.java

package CIE;

public class Student {

    public String usn;

    public String name;

    public int sem;

    public Student (String u, String n, int s) {

        this.usn=u;

        this.name=n;

        this.sem=s;

}

3

CIE/internals.java

package CIE;

public class Internals extends CIE.Student {

    public double imarks [];

    public Internals (String u, String n, int s, double m) {

        super(u, n, s);

        this.imarks = m;

3

## SEE/externals.java

```
package SEE;
import CIF.student;
```

```
public class externals extends CIF.Student {
    public double marks[]; // marks of 5 subjects
    public externals(String n, String r, int d) {
        super(n, r, d);
        this.marks = m;
    }
}
```

## result/test.java

```
package result;
import CIF.student;
import CIF.internals;
import SEE.externals;
public class test {
    public static void main (String [] args) {
        double internal[] = {43, 45, 47, 44, 41};
        double external[] = {90, 87, 65, 98, 73};
        student s1 = new student ("1bm22cs166", "Rukesh", 3);
        internals i1 = new internals ("1bm22cs166", "Rukesh", 3, "internal");
        externals e1 = new externals ("1bm22cs166", "Rukesh", 3, "external");
        System.out.println ("us: " + s1.usn + " name: " + s1.name);
        System.out.println ("internal marks: ");
        for (int i=0; i<5; i++) {
            System.out.printf ("internal marks %d: %f\n", i+1, i1.marks(i));
        }
        System.out.println ("external marks");
    }
}
```

```

for( int i=0; i<18; i++ ) {
    System.out.print( "*****.d:\\" + i + " " );
}

```

3

Output

CD n : IBM 22CS 166 name : Meikurd sem : 3

internal marks :

internal marks 1 : 43

" 2 45

3 47

4 44

5 49

external marks

external marks :

external marks 1 : 98

2 : 87

3 : 65

4 : 98

" 3 );

Kurd" , 3 , Meikurd ),

Meikurd , 3 , external ),

81. name "sem" );

it1 , it1 . marks ( 7 );

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.

```
class NT implements Runnable{
    Thread t;
    NT() {
        t = new Thread(this , "child thread");
        System.out.println("Child thread: "+t);
        t.start();
    }
    public void run() {
        try {
            for(int i=5 ; i>0 ; i--) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        }
        catch (InterruptedException e) {
            System.out.println("main thread interrupted");
        }
    }
}
class lab_5_2 {
    public static void main(String[] args) {
        new NT();
        try {
            for(int i=5 ; i>0 ; i--) {
                System.out.println("BMS college of Engineering");
                Thread.sleep(10000);
            }
        }
        catch (InterruptedException e) {
            System.out.println("main thread interrupted");
        }
    }
}
```

---

Child thread: Thread[ #29, child thread, 5, main ]

BMS college of Engineering

CSE

CSE

CSE

CSE

CSE

BMS college of Engineering

BMS college of Engineering

BMS college of Engineering

BMS college of Engineering

Write a program that creates two threads, one thread displaying BMS College of engg one in 10 seconds and another thread which displays CSE at 2 seconds

Class NT implements Runnable &  
Thread t;

NT() {

t = new Thread (this, "child thread");  
SOP ("child thread" + t);  
t.start();

}

public void run() {

try {

for (int i = 5; i > 0; i--) {

SOP ("CSE");

Thread.sleep (2000);

}

}

catch (InterruptedException e) {

SOP ("main thread interrupted")

}

} 3

```
public class lab_5_2 {  
    public static void main(String[] args)  
    {  
        new NTC();  
        try {  
            for (int i = 5; i > 0; i--) {  
                SOPC("BMS College of Engg.");  
                Thread.sleep(1000);  
            }  
        } catch (InterruptedException e) {  
            SOPC("main thread interrupted");  
        }  
    }  
}
```

Output

BMS College of Engineering

CSE

CSE

CSE

CSE

BMS College of Engineering

BMS College of Engineering

BMS College of Engineering

BMS College of Engineering

9/2

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.*;
import java.awt.event.*;

public class DivisionCalculator extends Frame implements ActionListener {
    private TextField num1Field, num2Field, resultField;

    public DivisionCalculator() {
        setTitle("Integer Division Calculator");
        setSize(300, 200);
        setLayout(new FlowLayout());

        Label num1Label = new Label("Num1: ");
        num1Field = new TextField(10);
        add(num1Label);
        add(num1Field);

        Label num2Label = new Label("Num2: ");
        num2Field = new TextField(10);
        add(num2Label);
        add(num2Field);

        Button divideButton = new Button("Divide");
        add(divideButton);
        divideButton.addActionListener(this);

        Label resultLabel = new Label("Result: ");
        resultField = new TextField(10);
        resultField.setEditable(false);
        add(resultLabel);
        add(resultField);

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent windowEvent) {
                System.exit(0);
            }
        });
    }

    public void actionPerformed(ActionEvent e) {
        try {
            int num1 = Integer.parseInt(num1Field.getText());
            int num2 = Integer.parseInt(num2Field.getText());
            if (num2 == 0) {
                throw new ArithmeticException("Cannot divide by zero");
            }
            int result = num1 / num2;
            resultField.setText(String.valueOf(result));
        } catch (NumberFormatException ex) {
            showMessageDialog("NumberFormatException", "Please enter valid integers for Num1 and Num2.");
        } catch (ArithmeticException ex) {
            showMessageDialog("ArithmeticException", "Cannot divide by zero.");
        }
    }
}

private void showMessageDialog(String title, String message) {
    Dialog dialog = new Dialog(this, title, true);
    dialog.setLayout(new FlowLayout());
    dialog.add(new Label(message));
    Button okButton = new Button("OK");
    okButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            dialog.dispose();
        }
    });
    dialog.add(okButton);
    dialog.setSize(200, 100);
    dialog.setVisible(true);
}

public static void main(String[] args) {
    DivisionCalculator calculator = new DivisionCalculator();
    calculator.setVisible(true);
}
```



DivisionOfIntegers



Number 1:

Number 2:

RESULT

Result: 24 8 3.0