

# Lab - 1

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

question

```

import java.util.Scanner;
import java.lang.Math;
public class QuadEqn {
    public static void main(String[] args) {
        Scanner s1 = new Scanner(System.in);
        float a = s1.nextFloat();
        float b = s1.nextFloat();
        float c = s1.nextFloat();
        float r1, r2;
        float d = (float) Math.sqrt(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 + " r2 = " + r2);
        } else if (d == 0) {
            r1 = -b / (2*a);
            System.out.println("r1 = " + r1 + " r2 = " + r2);
        } else {
            r1 = (float) Math.sqrt(-d) / (2*a);
            r2 = -1 * r1;
        }
    }
}

```

~~Output~~ ~~Java program~~ ~~Java code~~ ~~Java program~~

~~1. Enter a number:~~ ~~1~~ ~~2~~  
~~-5~~ ~~1~~ ~~2~~  
~~6~~ ~~1~~ ~~2~~ ~~3~~  
~~r1 = 3.0 r2 = 2.0~~ ~~r1 = 0.5 - 0.785i~~ ~~r1 = -1 r2 = -1~~  
~~-0.5 + 0.785i~~

## Lab - 2

66

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 ;

3

public string return\_usn () {

return this.usn ;

3

public string return\_name () {

return this.name ;

3

public int [ ] return\_credits () {

return this.credits ;

3

public float [ ] return\_marks () {

return this.marks .

3

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[i].credits = marks[i].credits +  
        credits[i] * marks[i];
```

return marks-credits / total-credits;

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

`Sgpa1 p = new Sgpa();`

P. Accept name cen ("Hukund," "Ibm22c12")  
int[7] 1991/18 3:15 54 x 4 2? = 1 13

int L] credits - 3rd = 4, 4, 4, 3, 3, 2, 1, 13  
P: accept - 100% ( 19 credits 3rd ) :

float [J] marks - 3rd = S > 3 + 1 , 45 20 |

for all 7 males:  $\Sigma d = 23.45$ ,  $45.78$ ,  
 $1.23$ ,  $87.61$ ,  $9.01$ ,  $65.43$ ,

125, 87.6, 9.01, 65.43, 1,  
34.86, 78.13;

q. accept marks (marks - 3rd);

System.out.println("User: " + p.getAnswer());

System-Dict geïmporteert ("name: " + p.return\_value)

~~int [ ] get\_credits = p.return\_credits;~~

```
for (int i=0; i<8; i++) {
```

```
System.out.print("marks: ", get_marks);
```

Enlarged with all

`System.out.println();`  
输出结果是什么?

psgpa (get credits, get marks).

~~marks~~ ~~for~~ Output

name: Plekland.

Name: Plekand.

user: tpm22 cs166

marks: 23.45, 14.45, 45

sepa: 3.71

- (2) Create a book class and create a constructor, get set and set methods and make a method for creating 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.numPage = np;
        this.price = p;
    }
    public String toString() {
        return "Name:" + name + " Author:" +
            author + " Price:" + price + " noPage:" +
            numPage;
    }
}
```

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.print("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;

```

```

9.
public static void main(String[] args) {
    int n;
    System.out.println("Mukund, 1 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 i = 0; i < n; i++) {
            System.out.println(b[i]);
        }
    }
}

```

3.

Output

Enter no. of books

# 1

Enter book name

Berserk

Enter author name

Kentaro Miura

Enter price

1000

Enter number of pages

200

Enter bookname

Name : Berserk

author : Kentaro Miura

price : 1000

number of pages : 100

1000/- per book

09/80

## Lab-3

66  
Page No. \_\_\_\_\_  
Date: \_\_\_\_\_, 99

- (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().

Soh abstract class shape {

    int x=0, y=0;

    public shape (int x, int y) {

        this.x = x;

        this.y = y;

    abstract void printArea();

    }

    public rect (int l, int b) {

        super (l, b);

    public void printArea() {

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

    }

    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 of Circle is " + pi * 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);
```

```
r1.printarea();
```

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

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

Output

area of rectangle: 60

area of triangle: 25.0

area of circle: 50.24

## 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

3

## SEE/externals.java

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

```
public class externals extends CIF.student {
    public double smarks[]; // marks array
    public externals(String u, String n, int d) {
        super(u, n, d);
        this.smarks = 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 ("1bm22cs16", "Hukund", 3);
        internals i1 = new internals ("1bm22cs16", "Hukund", 3, internal);
        externals e1 = new externals ("1bm22cs16", "Hukund", 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 ("%d", i1.i1.marks[i]);
        }
        System.out.println ("external marks:");
    }
}
```

```

for( int i=0; i<18; i++ ) {
    System.out.print("externally: " + j + " ");
}
    
```

3

Output

as in : IBM 22CS 166 names: Meekund 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

2, 3);

Kund, 3, Meekund);

Meekund, 3, external);

SI. name sem: 3;

it1, it1.marks(i));

(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;

}

3

class Son extends Father {

int age;

public Son extends Father {

super.setAge();

this.age = sonAge;

if (this.age > super.age)

throw 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("\_\_\_\_\_");

}

Output (Ans)

Age cannot be lesser than given age is: 0

Please enter again.

Exception caught: Exception: 0

# Lab - 6

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

①

## Exception

class SalaryException extends Exception {

public SalaryException (String msg) {  
System.out.println ("Entered");

3

class Manager {

int salm;

public Manager (int sal) throws SalaryException {  
if (sal < 0) throw SalaryException ("Sal not  
this.salm = sal;

3

class Worker extends Manager {

int salw;

public Worker (int salc, int salm) throws  
Super (salm);

if (salw > salm) throw new SalaryException  
this.salw = salw;

3

public class Employee

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

Manager m1 = new Manager (1)

Worker w1 = new Worker (32000)

3

Catch (SalaryException e) {

System.out.println("Exp caught "+e);

}

}

Output

entered sal is wrong: salary cannot be lesser zero  
 Exception caught  
 Salary Exception

②

Generics

class Gen<T> {

T ob;

Gen<T> o) { ob = o; }

T getob() { return ob; }

void showType() {

System.out.println("Type of T is "+ );

}

public class GenericsTest {

public static void main(String[] args) {

Gen<Integer> lob = new Gen<Integer>(88);

System.out.println(lob.getob());

lob.showType();

Gen<String> job2 = new Gen<String>(mySt);  
System.out.println(job2.getObj());  
job2.showType();

Gen<Boolean> job3 = new Gen<Boolean>(true);  
System.out.println(job3.getObj());  
job3.showType();

3  
3

Output

88

Type of T is java.lang.Integer

This is my string

Type of T is java.lang.String

true

Type of T is java.lang.Boolean

QW  
29/11/2023

# Lab - 7

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

## → Sample programs

class Thread1 {

```
public static void main(String [] args){  
    Thread t = Thread.currentThread();  
    System.out.println("CT : " + t);  
    t.setName("Current main Thread");  
    System.out.println("CT : " + t);
```

for {

```
for (int n=5; n>0; n--) {  
    System.out.println(n);  
    Thread.sleep(500);
```

catch (InterruptedException ie) {  
 System.out.println("break");

}

output

5

4

3

2

1

class NewThread implements Runnable {

    Thread t;

    NewThread() {

```
        t = new Thread(this, "NThread");
        System.out.println("CT : " + t);
        t.start();
```

}

    public void run() {

        try {

```
            for(int n=5; n>0; n--) {
                System.out.println(n);
```

}

    } catch(InterruptedException ie) {

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

}

```
        System.out.println(" quitting CT ");
```

}

class Thread2 {

    public static void main(String[] args)

        new NewThread().

        System.out.println("Back in main")

        try {

```
            for(int n=5; n>0; n--) {
```

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

}

        } catch(InterruptedException ie) {

```
            System.out.println(" quitting MT ");
```

}

}

Output

Back in main.

child : 5

Main thread : 5

child : 4

Main thread : 4

child : 3

child : 2

Main thread : 3

child : 1

Main : 2

Write a program that creates two threads, one thread displaying BMS College of engg once 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--) {

            SOP ("BMS college of engg.");  
            Thread.sleep (10000);

}

}

    catch (InterruptedException e) {

        SOP ("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

Q 6/2

## 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;
```

3

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

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

3

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

3

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

```
        else {
```

```
            balance -= amount;
```

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

3

3

class Sav-Acc extends Account {  
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")

3

## public class Bank {

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

Sav-aect savings = new Sav-aect (T, 123),

Cur-aect current new Cur-aect CTD,<sup>98</sup>

```
System.out.println("Enter amt : ");
```

depositAmount = s.nextDouble();

Current Amount • deposit CdepositPlan

Current Account - cheques & post  
current Account. display Balance();

```
System.out.print("enter aint:");
```

withdrawAmount = s. nextDouble( ).

current account. withdraw (withdrawal);

Current Account - displayBalance();

long and

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

Double withdrawal Amount = S.nextDouble()

Savings Account. withdrawl (withdrawals);

Carry Account. display Balance ( ),

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

authdeanAmount = scanner. not doubly

current Account, cashflow (without bond),

currentAccount.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 : 280000

Interest deposited : 998.0

Current Balance : 20958