

# Krishna Mehta(1BM22CS130)

**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");
    }
}
}

```

Q. 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;
Class Quadratic
{
    int a, b, c;
    double r1, r2, d;
    void getd()
    {
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter the coefficients of
a, b, c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }
    void compute()
    {
        while (a == 0)
        {
            System.out.println ("Not a quadratic equation");
            System.out.println ("Enter a nonzero value of a");
            Scanner s = new Scanner (System.in);
            a = s.nextInt(); }
    }
    d = b*b - 4*a*c;
    if (d == 0)
    {
        r1 = (-b) / (2*a);
        System.out.println ("Roots are real & equal");
    }
}

```

```

    System.out.println ("Root1 = Root2 = " + n1);
}
else if (d > 0)
{
    n1 = ((-b) + (Math.sqrt(d)))/ (double) (2*a);
    n2 = ((-b) - (Math.sqrt(d)))/ (double) (2*a);
    System.out.println ("Roots are real & distinct");
    System.out.println ("Root1 = " + n1 + "Root2 = " + n2);
}
else (d < 0)
{
    System.out.println ("Roots are imaginary & real roots do not exist");
}
}

```

```

class Quadratic
{
    public static void main (String xx[])
    {

```

```

        Quadratic q = new Quadratic ();
        q.getd();
        q.compute();
    }
}

```

=> Output :

Enter the Coefficients a,b,c

1

2

-8

Roots are real & distinct  
 Root1 = 2.0      Root2 = -4.0

**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 progm 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 progm 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 progm 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);
    }
}
```

Q:

1-1-24

PAGE NO.:  
DATE:

Q- 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.

~~Print");  
+ s2);}~~

~~→ import java.util.Scanner;  
class Book {  
String name, Author;  
int price, numPages;~~

~~My %~~

```
import java.util.Scanner;
class Student {
    String USN, Name;
    Scanner SI = new Scanner (System.in);
    float credits [] = new float [8];
    float marks [] = new float [8];
    void accept ()
    { System.out.println ("Enter the USN & Name :");
        USN = SI.next();
        name = SI.next();
        System.out.println ("Marks in 1: Maths
2. Physics 3. C prog 4. Web 5. Kannada 6: IDT
7. Civil 8. English");
        for (int i=0; i<8; i++)
        { System.out.print ((i+1) + " = ");
            marks [i] = SI.nextInt(); }
        System.out.println ("Respective credits of subjects
1. Maths 2. Physics 3. C prog 4. Web
5. Kannada 6. IGT 7. Civil 8. English");
        for (int i=0; i<8; i++)
        { S.O.P ((i+1) + " = "); credits [i] = SI.nextInt(); }
```

```

void display()
{
    S.O.P ("USN : " + USN + " name : " + name);
    S.O.P ("Marks and credits in the following
    subjects are 1. Maths ..... 8. English");
    for (int i=0; i<8; i++)
    {
        S.O.P ((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] <= 49) return 6;
    else if (marks[i] >= 40 && marks[i] <= 39) return 5;
    else if (marks[i] >= 33 && marks[i] <= 39) return 4;
    else return 0;
}

```

```

float sgpa()
{
    float SGPA, sum=0, sum2=0;
    for (int i=0; i<8; i++)
    {
        sum = sum + gpa (i) * credits[i];
    }
    for (int i=0; i<8; i++)
    {
        sum2 = sum2 + credits[i];
    }
    float tc = sum2;
    SGPA = sum / sum2;
    return SGPA;
}

```

```
class test {  
public static void main (String [] args)  
{ student sl = new student ();  
sl.accept ();  
sl.display ();  
float Ans ;  
Ans = sl.sgpac ();  
S.O.P ("SGPA is : " + Ans );  
}}
```

=> Output :

Enter the VSN and Name:

113

Krishna

Marks in following Subjects 1. Maths ... 8. English :

1 = 99

2 = 54

3 = 21

4 = 54

5 = 36

6 = 95

7 = 78

8 = 45

Respective Credits in 1. Maths ... 8. English

1 = 4

2 = 4

3 = 3

4 = 3

5 = 1

6 = 1

7 = 3

8 = 1

**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_goggins", 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();
}
}

```

Q2. 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 objects. Include (toString()) method that could display the complete details of the book. Create n objects.

→ import java.util.Scanner;  
class Book {  
String name, Author;  
int price, num-pages;  
Book () {};  
void set () {  
S.O.P ("Enter details of book :");  
Scanner SI = new Scanner (System.in);  
S.O.P ("Enter name :");  
name = SI.next();  
S.O.P ("Enter Author's name.");  
Author = SI.next();  
S.O.P ("Enter price :")  
price = SI.nextInt();  
S.O.P ("Enter No. of pages :");  
num-pages = SI.nextInt();  
}
}

PAGE NO. \_\_\_\_\_  
DATE \_\_\_\_\_

```

void get()
{
    System.out.println("name:" + name + " Author:"
+ Author + " price:" + price + " num-pages:"
+ numPages);
}

class bookdemo {
    public static void main (String [] args)
    {
        int n; i=0;
        Scanner S = new Scanner (System.in);
        System.out.print("n:");
        n = S.nextInt();
        Book B[] = new Book [n];
        for (i=0 ; i < n ; i++)
        {
            B[i] = new Book();
            B[i].set();
            B[i].get();
        }
    }
}

```

=) Output:

~~X~~ 11/24

~~Print n: 2~~

~~Enter details of book:~~

~~Name : Journey~~

~~Author: Krishna~~

~~Price: 999~~

~~num-pages: 700~~

~~name: Journey Author: Krishna Price: 999 num.: 700~~

~~Enter details of book:~~

~~Name: lost~~

~~Author: Elve~~

~~Price: 500 num - pages = 50~~

~~name: lost Author: Elve Price: 500 num-pages: 50~~

**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();  
}  
}
```

8-1-24

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 class  
the  
Area()  
shape.

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

✓ 814.0

• ps :

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:
```

```
        System.out.print("Enter amount to withdraw: ");
        double withdrawAmount = scanner.nextDouble();
        System.out.print("Select account (1 for Savings, 2 for
Current): ");
        accountChoice = scanner.nextInt();
        if (accountChoice == 1)
            savings.withdraw(withdrawAmount);
        else if (accountChoice == 2) {
            current.withdraw(withdrawAmount);
            current.deductServiceCharge();
        }
        break;
    case 3:
        System.out.print("Select account (1 for Savings, 2 for
Current): ");
        accountChoice = scanner.nextInt();
        if (accountChoice == 1)
            savings.displayBalance();
        else if (accountChoice == 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.");
    }
}
}
```

## Java

### ) LAB - Program - 5

Q. Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account & the other current account. The savings account provides compound interest & 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 & if the balance fall below, a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the classes Curr-acct & Sav-acct to make them more specific to their requirements. Include the necessary methods:

- a] Accept deposit from customer & update balance
- b] Display balance
- c] Compute & deposit interest
- d] Permit withdrawal & update the balance.

-) import java.util.Scanner;

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

PAGE NO.:  
DATE:

```

class Bank
for its
+ & the
ccount
al
The
ability
lders
lance
ice
nts.
nce

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 Rs." + amount
                        + " successful.");
}

void displayBalance ()
{
    S.O.P ("Balance: Rs." + balance);
}

class CurrAcct extends Account
{
    double minBalance;
    double serviceCharge;
    CurrAcct (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;
            S.O.P ("Withdrawal of Rs." + amount +
                   " Successful.");
        }
        else S.O.P ("Insufficient Funds.");
    }
}

```

```
PAGE NO. _____  
DATE: _____  
void deductServiceCharge()  
{ if (balance < minBalance)  
{ balance -= serviceCharge;  
S.O.P ("Service charge of Rs." +  
serviceCharge + " applied due to  
balance below minimum."); } }
```

```
class SavAcct extends Account  
{ double interestRate;  
SavAcct (String name, int accNo, String accType,  
double initialBalance, double interest)  
{ interestRate = interest; }}
```

```
void calculateInterest()  
{ double interest = balance * interestRate / 100;  
balance += interest;  
S.O.P ("Interest of Rs." + interest + " added."); }
```

```
void withdraw (double amount)  
{ if (balance - amount >= 0)  
{ balance -= amount;  
System.out.println ("Withdrawal of Rs."  
+ amount + " successful."); }  
else {  
S.O.P ("Insufficient funds."); } }
```

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

```
SavAcct savings = new SavAcct ("Krishna",  
123456, "Savings", 1000, 5);
```

Cu & Acct Current = new CurAcct ("Krishna",  
12345678, "Current", 2000, 500, 10);

S.O.P ("Welcome to our Bank!");

while (true)

{ S.O.P ("In 1. Deposit In 2. Withdraw  
In 3. Display Balance In 4. Exit");

S.O.P ("Enter your choice: ");

int choice = sc.nextInt();

switch (choice)

{ case 1:

S.O.P ("Enter amount to deposit: ");

double depositAmount = sc.nextDouble();

S.O.P ("Select account (1 for Saving, 2 for  
Current): ");

int accountChoice = sc.nextInt();

if (accountChoice == 1)

Savings.deposit (depositAmount);

else if (accountChoice == 2)

current.deposit (depositAmount);

break;

Case 2:

S.O.P ("Enter amount to withdraw: ");

double withdrawAmount = sc.nextDouble();

S.O.P ("Select Account (1 for saving, 2 for  
Current): ");

accountChoice = scanner.nextInt();

if (accountChoice == 1)

Savings.withdraw (withdrawAmount);

```
PAGE NO.:  
DATE:  
  
else if (accountChoice == 2)  
{ current.withdraw ( withdrawAmount );  
current.deductServiceCharge (); }  
break;  
Case 3:  
S.O.P ("Select account (1 for Saving, 2 for  
Current): ");  
accountChoice = scanner.nextInt();  
if (accountChoice == 1 )  
    savings.displayBalance ();  
else if (accountChoice == 2 )  
    current.displayBalance ();  
break;  
Case 4:  
S.O.P ("Thank You");  
System.exit (0);  
default:  
    S.O.P ("Invalid choice"); } } }
```

=> Output:

Welcome to our Bank!

- 1. Deposit
- 2. Withdraw
- 3. Display Balance
- 4. Exit

Enter your choice: 1.

Enter amount to deposit: 8000

Select account (1 for Savings, 2 for Current): 1

Enter your choice: 1.

AMOUNT  
DATE

Enter amount to deposit: 1500

Select account (1 for Savings, 2 for Current): 2

Enter choice: 3

Select account: 1

Balance: Rs. 3000.0 -  
for

Enter choice: 3

Select account: 2

Balance: Rs. 3500.0 -

Enter choice: 2

Select account: 1 Enter amount 2000

Balance: Withdrawal of Rs. 2000.0 successful.

Enter choice: 4 -

STIM

1:

**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].getcie();
            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();
}
}

```

Java LAB

Q-6- 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.

```

1 =) Package CIE;
import java.util.Scanner;
public class Student {
    public String USN;
    public String Name;
    public int Sem;
    Scanner s = new Scanner (System.in);
    public void accept () {
        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 + "\nUSN:" +
this.USN + "\nSem:" + this.sem);
    }
}

```

=) Package CIE;  
import java.util.Scanner;  
public class Internal extends CIE.Student  
{ public int m [] = new int [5];  
CIE.Student student = new CIE.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]); } }.

=) Package SEE;  
import java.util.Scanner;  
import CIE.Internal;  
import CIE.Student;  
public class External extends CIE.Student  
{ public int x [] = new int [5];  
public void accept()  
{ Scanner s2 = new Scanner (System.in);  
System.out.println ("Enter External Marks:");  
for (int i=0; i<5; i++)  
{ x [i] = s2.nextInt(); } }  
public void display ()  
{ super.display ();  
for (int i=0; i<5; i++)  
{ System.out.println ("Marks of sub" + (i+1) + " = " + x [i]); } } }.

```

⇒ import java.util.Scanner;
import CIE.Student;
import CIE.Internal;
import SEE.External;
public class Final {
    public static void main (String [] args) {
        Scanner n = new Scanner (System.in);
        System.out.println ("Enter n:");
        int y = n.nextInt ();
        CIE.Internal [] c1 = new CIE.Internal [y];
        SEE.External [] c2 = new SEE.External [y];
        for (int i=0; i<y; i++) {
            c1 [i] = new CIE.Internal ();
            c2 [i] = new SEE.External ();
            c1 [i].accept ();
            c2 [i].accept ();
            c1 [i].display ();
            c2 [i].display ();
        }
    }
}

```

```

for (int j=0; j<5; j++)
{
    double calc = c1 [j].m [j] + ((c2 [j]+x [j])/2);
    System.out.println ("Final marks of sub"
                        + (j+1) + "]= " + calc);
}

```

⇒ Output :

Enter n: 2

Enter Name: abc

Enter USN: 10

Enter Sem: 2

Enter Internal Marks:

40

41

42

43

44

Enter Ext marks:

50 49 47 46 44.

=) Name: A

USN: 10

Sem: 2

Marks of Sub1 = 40

Marks of Sub2 = 41

Marks of Sub3 = 42

Marks of Sub4 = 43

Marks of Sub5 = 44

=) Name: B

USN: 11

Sem: 2.

Marks of Sub1 = 50

Marks of Sub2 = 49

Marks of Sub3 = 47

Marks of Sub4 = 46

Marks of Sub5 = 44

Marks of Sub1 = 143

Marks of sub2 = 83

Marks of sub3 = C

Marks of sub4 = 2

Marks of sub5 = 2

Final Marks of sub[1] = 172

Final Marks of sub[2] = 92

Final Marks of sub[3] = 46

Final Marks of sub[4] = 99

Final Marks of sub[5] = 80

**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());
    }
}
}

```

Q- 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 takes both father and son's age and throws an exception if son's age is  $\geq$  father's age.

$\Rightarrow$  import java.util.Scanner;  
class WrongAgeException extends Exception  
{  
 public 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),

handling  
Create a  
red class  
class.  
ctor which  
ption  
In  
sturct  
's age  
 $>=$  father's age  
message)

Pb (sonAge  $\geq$  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)  
{ Scanner s = new Scanner (System.in);  
try {

S.O.P ("Enter Father's Age :");

int fatherAge = s.nextInt();

Father f = new Father (fatherAge);

S.O.P ("Enter son's Age :");

int sonAge = s.nextInt();

Son son = new Son (fatherAge, sonAge);

}

catch (AgeException e)

{ S.O.P ("Exception: " + e.getMessage());  
}

=> Output:

Enter's Father's Age: 25

Enter's son's Age: 14.

Father's age = 25

Son's age = 14.

**Q8.** Write a program to create a 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();
    }
}
```

## Java

### Threads

```
=> class NewThread1 implements Runnable  
{ Thread t1;  
NewThread1()  
{ t1 = new Thread(this, "Thread1");  
System.out.println("CT: " + t1);  
t1.start();  
}  
public void run()  
{ try  
{ for (int n=5; n>0; n--)  
System.out.println("BMS College of  
Engineering");  
Thread.sleep(10000);  
}  
catch (InterruptedException ie)  
{ System.out.println("Thread1 interrupted");  
}  
System.out.println("Thread1 quitting");  
}
```

~~```
class NewThread2 implements Runnable  
{ Thread t2;  
NewThread2()  
{ t2 = new Thread(this, "Thread2");  
System.out.println("CT: " + t2);  
t2.start(); }  
public void run()  
{ try  
{ for (int n=5; n>0; n--)  
System.out.println("CSE");  
Thread.sleep(2000); } }
```~~

```
PAGE NO. _____  
DATE _____  
catch (InterruptedException ie)  
{ System.out.println ("Thread 2 is interrupted");  
}  
System.out.println ("Thread 2 is quitting");  
}  
}  
class Main Thread  
{ public static void main (String ss[]){  
new NewThread1 ();  
new NewThread2 ();  
}  
}
```

=) Output :-

CT: Thread [#24; Thread 1, S, main]

BMS College of Engineering

CT: Thread [#30, Thread 2, S, main]

CSE

CSE

pted");

CSE

CSE

CSE

BMS College of Engineering

Thread 2 quitting

BMS College of Engineering

BMS College of Engineering

BMS College of Engineering

Thread 1 quitting

**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 Arithmetic Exception 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
```

```
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 (ArithmetricException e) {
JOptionPane.showMessageDialog(this, "ArithmetricException: 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();
} });}}
```

Q. Write a Program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 & 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 NumberFormat Exception. If Num2 were 0, the program would throw Arithmetic Exception. Display the exception message in dialog box.

```
→ import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
public class Exception Demo extends JFrame
    implements ActionListener {
    private JTextField t1, t2, t3;
    private JLabel l1, l2;
    private JButton b1, b2;
    public Exception Demo () {
        setLayout (new FlowLayout ());
        l1 = new JLabel ("Num 1:");
        add (l1);
        t1 = new JTextField (5);
        add (t1);
        l2 = new JLabel ("Num 2:");
        add (l2);
        t2 = new JTextField (5);
        add (t2);
        t3 = new JTextField (5);
        t3.setEditable (false);
        add (t3);
        b1 = new JButton ("Divide");
    }
}
```

user interface  
user enters  
in 2 Num.  
displayed  
button is  
not an  
NumberFormat  
the program  
play the

```
add(b1);
b1.add ActionListener(this);
b2=new JButton ("Clean");
add(b2);
b2.addActionListener(this);
```

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(300, 150);
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 (ArithmaticException e)
{ JOptionPane.showMessageDialog(this, "ArithmaticException: Cannot divide by zero!"); }
```

```
catch (NumberFormatException e)
{ JOptionPane.showMessageDialog(this, "NumberFormatException: Please enter valid integers for num1 & num2."); }
```

```
else { t1.setText ("");
t2.setText ("");
t3.setText (""); }}
```

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

Output:

Num 1: 12

Num 2: 6 2

Divide

Clear

## Generics:

GENERICS

```
class Gener <G, T>
{
    G ob; T ob1;
    Gener (G o, T o1)
    { ob=o; ob1=o1; }

    void showtype()
    {
        System.out.println (ob.getClass().getName());
        System.out.println (ob1.getClass().getName());
    }

    G getobj()
    {
        return ob;
    }

    T getobj1()
    {
        return ob1;
    }
}

class GenMain
{
    public static void main (String args[])
    {
        Gener <Integer, Double> g1;
        g1=new Gener <Integer, Double> (100, 20.985);
        g1.showtype();

        int x = g1.getobj();
        System.out.println (x);
        double d= g1.getobj1();
        System.out.println (d);
    }
}
```

✓

=> Output:

java.lang.Integer  
java.lang.Double  
100  
20.985.

## Report For Applet Programs:

LAB-07

PAGE NO. \_\_\_\_\_  
DATE: \_\_\_\_\_

④ AWT Programs - REPORT

① Buttondrag.java : It opens a Button Game window with 3x3 tiles with numbers & provides user with buttons for reset , start and restart. Once clicked on start, user can click on two tiles and simultaneously clicked tiles get swapped. User can click on reset to start the game again.

② ButtonList.java : It opens a ButtonList window with three buttons - yes, no, undecided and has a default text HELLO. Once clicking on yes, the window writes the text = You pressed Yes. Similarly on clicking no → You pressed No. and on clicking undecided → You pressed undecided.

③ ButtonListD.java : It opens ButtonList D window with three buttons - yes, no, undecided. On clicking yes, a Dialog window appears with message You pressed Yes and an Ok button. Similarly on clicking No, You pressed No & on clicking undecided, you pressed undecided in separate dialog box.

X 19

- 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 window after Result: If gives & intakes numbers in float data type.
- ⑤ Division Main.java - It opens a Division of Integers window similar to last program. But the result provided will be integer numbers & the quotient in float.
- ⑥ TextField 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 entered appears after Name. Similarly on clicking enter after entering the password, it appears next to password. On selecting some letters in name field and clicking enter, the selected text appears next to Select text in name. Password receives the input & displays it after encrypting it.

✓  
18/2/19