

NAME : Tanya D Shetty STD : 3 B SEC : _____ ROLL NO : _____

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
1.	8/1/23	Program 1 - Student Details	1	10
2.	8/1/23	Program 2 - Quadratic Equation	2	10
3.	8/1/23	Program 3 - Books	3	10
4.	22/1/23	Program 4 Shape (Abstract)	4	10
5.	22/1/23	Program 5 - Account (current & savings)	5	10
6.	30/1/23	Program 6 - Packages	9	10
7.	19/2/23	Program 7 - Exception Handling	10	18
8.	19/2/23	Program 8 - Threads	12	10
9.	26/2/23	Program 9 - AWT (Frame)	14	
10.	26/2/23	Program 10 - AWT (Event Handling)	15	

Program - 1

1) student details

(Marksheet main [1] to
(11.11.2018) 12

```
import java.util.Scanner;  
class student  
{  
    int usn;  
    String st_name = new String();  
    double marks[] = new double[7];
```

void getd()

```
{  
    System.out.println("Enter Student details:");  
    Scanner ss1 = new Scanner(System.in);  
    usn = ss1.nextInt();  
    st_name = ss1.next();  
    for (int i = 0; i < 6; i++)  
    {  
        marks[i] = ss1.nextDouble();  
    }  
}
```

void putd()

```
{  
    System.out.println("Student details:");  
    System.out.println(usn + " " + st_name);  
    for (int i = 0; i < 6; i++)  
    {  
        System.out.print(marks[i] + " ");  
    }  
}
```

class Run

```
{  
    public static void main (String s[])  
    {  
        System.out.println("Enter no. of students:");  
        Scanner ss2 = new Scanner(System.in);  
        int n = ss2.nextInt();  
        student s1[] = new student[n];  
        for (int i = 0; i < n; i++)  
        {  
            s1[i].getd();  
            s1[i].putd();  
        }  
    }  
}
```


Quadratic Equation

```
import java.util.Scanner;  
import java.lang.Math;  
class Quadratic {  
    int a, b, c;  
    double d;  
    double r1, r2;  
    void get (int a, int b, int c) {  
        this.a = a;  
        this.b = b;  
        this.c = c;  
    }  
    void calculate () {  
        this.d = b * b - 4 * a * c;  
        if (d > 0) {  
            System.out.println ("quadratic equation has real & diff roots");  
            r1 = (-b + Math.sqrt(d)) / (2 * a);  
            r2 = (b + Math.sqrt(d)) / (2 * a);  
            System.out.println ("the roots are: " + r1 + " & " + r2);  
        } else if (d == 0.0) {  
            System.out.println ("quadratic equation has real & equal roots");  
            r1 = (-b) / (2 * a);  
            r2 = r1;  
            System.out.println ("quadratic roots are: " + r1 + " & " + r1);  
        } else {  
            System.out.println ("quadratic equation has imaginary roots");  
        }  
    }  
}  
class Quad {  
    public static void main (String args []) {  
    }  
}
```

```

Quadratic q = new Quadratic();
System.out.println("Enter value of a,b,c");
Scanner s = new Scanner(System.in);
q.a = s.nextInt();
q.b = s.nextInt();
q.c = s.nextInt();
q.get(q.a, q.b, q.c);
q.calculate();
}
}

```

O/P ①

Enter value of a,b,c

1 2 1

quadratic equation has real and equal roots
the roots are: -1.0 & -1.0

O/P ②

Enter value of a,b,c

1 1 1

quadratic equation has imaginary roots

(0.0±b)

O/P ③

Enter value of a,b,c

1 6 3

quadratic equation has real and diff. roots,
the roots are: -0.55051, -5.44949

Books

```
import java.util.Scanner;  
class Books  
{  
    String name;  
    String author;  
    int price;  
    int numPages;  
    Books() {}  
    Books(String name, String author, int price, int numPages)  
    {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.numPages = numPages;  
    }  
    public String toString()  
    {  
        String name, author, price, numPages;  
        name = "Book name:" + this.name + "\n";  
        author = "Author name:" + this.author + "\n";  
        price = "Price:" + this.price + "\n";  
        numPages = "number of pages:" + this.numPages + "\n";  
        return name + author + price + numPages;  
    }  
}
```

class Main

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

```

int numPages;
System.out.println("Enter number of books: ");
n = s.nextInt();
Books b[] = new Books[n];
for (int i = 0; i < n; i++) {
    System.out.println("Book " + (i + 1) + ":");
    System.out.print("Enter name of book: ");
    name = s.next();
    System.out.print("Enter author: ");
    author = s.next();
    System.out.print("Enter author: ");
    author = s.next();
    System.out.print("Enter price: ");
    price = s.nextInt();
    System.out.print("Enter no of pages: ");
    numPages = s.nextInt();
    b[i] = new Books(name, author, price, numPages);
}
for (int i = 0; i < n; i++) {
    System.out.println("Book " + (i + 1) + ": " + b[i]);
}

```

OUTPUT:

Enter the number of books: 2

Book 1:

Enter the name of book: Jungle Book

Enter author of book: R Kipling

Enter the price of book: 1000

Enter number of pages of book: 500

Book 2:

Enter the name of book: Matilda

Enter the author of book: Roald Dahl

Enter the price of book: 100

Enter the number of pages of book: 300

Book 1:

Book name : Jungle Book

Author name: R Kipling

Price : 1000

Number of pages: 500

Book 2:

Book Name: Matilda

Author name : Roald Dahl

Price : 900

Number of pages : 300

Done
8/1/24

④ Shapes Class (to find area - abstract class)

```
import java.util.*;  
abstract class shape {  
    float l, b, r;  
    Scanner input = new Scanner(System.in);  
    abstract protected void printArea();  
}
```

class Rectangle extends shape {

```
    void printArea() {  
        System.out.println("Find area of rectangle:");  
        System.out.println("Enter length and breadth:");  
        l = input.nextInt();  
        b = input.nextInt();  
        System.out.println("area=" + l * b);  
    }  
}
```

class Triangle extends shape {

```
    void printArea() {  
        System.out.println("Find area of triangle:");  
        System.out.println("Enter height and base:");  
        l = input.nextInt();  
        b = input.nextInt();  
        System.out.println("area=" + 0.5 * l * b);  
    }  
}
```

class Circle extends shape {

```
    void printArea() {  
        System.out.println("Find area of circle:");  
        System.out.println("Enter radius:");  
        r = input.nextInt();  
        System.out.println("area=" + 3.142 * r * r);  
    }  
}
```

class Run {

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

Rectangle a = new
Triangle b = new
Circle c = new

- a. printArea();
- b. periArea();
- c. printArea();

?

?

OUTPUT:

Find area of rectangle

Enter length and breadth: 10 2

Area = 20 [0.074 141666 pi * r * r] atring. two. output

Find area of triangle

Enter height and base: 30 40

Area = 600 [0.074 141666 pi * r * r] atring. two. output

Find area of circle

Enter radius: 40

Area of circle = 50.24.0

```
import
```

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

```
class Account
```

```
{
```

```
    public static int min=500;
```

```
    String name;
```

```
    int Acno;
```

```
    public float o_Price;
```

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

```
    public void get_info()
```

```
{
```

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

```
        name = input.nextLine();
```

```
        System.out.println ("Enter Account Number");
```

```
        Acno = input.nextInt();
```

```
        System.out.println ("Enter opening amount >500:");
```

```
        o_Price = input.nextFloat();
```

```
        if (o_Price < 500)
```

```
{
```

```
            System.out.println ("Enter opening amount must >500.");
```

```
}
```

```
}
```

```
    public void show()
```

```
{
```

```
        System.out.println ("Name " + name);
```

```
        System.out.println ("Acno " + Acno);
```

```
        System.out.println ("Amount " + o_Price);
```

```
}
```

```
}
```

```
class Current extends Account
```

```
{
```

```
    float deposit , withdraw , penalty;
```

```
    public void deposit()
```

```
{
```

```
        System.out.println ("Enter Amount to deposit");
```

```
        deposit = input.nextFloat();
```

```
        show();
```

$O_Price = O_Price + deposit;$
System.out.println("Total Amount is " + O_Price);

}

```
public void check_Bal()
```

```
{ if (O_Price < min)
```

System.out.println("Amount should > 500");

O_Price = O_Price - 150;

System.out.println("You have debited amount 150 from
your ^{new} balance is " + O_Price);

}

```
public void withdraw_Bal()
```

```
{ System.out.println("Enter amount to withdraw");
```

withdraw = input.nextFloat();

show();

```
/if (O_Price < 500)
```

```
{ System.out.println("For withdrawal, balance must > 500");
```

}/

```
if (withdraw < O_Price)
```

{

O_Price = O_Price - withdraw;

System.out.println("Withdrawn Balance " + O_Price);

}/

else

{

System.out.println("Insufficient Balance cannot be < 500");

}/

check_Bal();

}/

}/

class Saving extends Account

```
{ float deposit, withdraw, interest;
```

```
public void deposit()
```

{

System.out.println("Enter amount to deposit");

deposit = input.nextDouble();

show();

o_Price = o_Price + deposit;

System.out.println("Total Amount is:" + o_Price);

}

public void check_interest()

{ interest = (o_Price * 2) / 100;

o_Price = o_Price + interest;

System.out.println("Total Amount with interest is" + o_Price);

}

public void withdraw_Bal()

{ System.out.println("Enter Amount to withdraw");

withdraw = input.nextDouble();

show();

if (withdraw < o_Price)

o_Price = o_Price - withdraw;

System.out.println("After withdrawal Balance:" + o_Price);

}

else

System.out.println("Insufficient Balance!");

}

}

public class Account2

{

static String ch;

public static void main (String [] args)

{

int count = 0;

Scanner sc = new Scanner (System.in);

Current cur = new Current();

Savings sav = new Savings();

```

System.out.println ("Choose Account type:");
System.out.println ("Press c for Current Acc:");
System.out.println ("Press s for Saving Acc:");
ch = sc.nextInt();
if (ch.equalsIgnoreCase("c"))
{
    cu.getinfo();
    cu.checkBal();
    while (count != 4)
    {
        System.out.println ("1. display\n2. Deposit\n3. Withdraw\n4. Exit");
        System.out.println ("Enter Your Choice:");
        int cho = sc.nextInt();
        switch (cho)
        {
            case 1: cu.show();
            break;
            case 2: cu.deposit(); // add growth along his method
            break;
            case 3: cu.withdraw();
            break;
            case 4: System.exit(0);
            break;
            default: System.out.println ("Wrong Choice");
        }
    }
}
else if (ch.equalsIgnoreCase("s"))
{
    sav.getinfo();
    while (count != 5)
    {
        System.out.println ("1. display\n2. Deposit\n3. Withdraw\n4. Exit");
        System.out.println ("Enter Your Choice:");
        int cho = nextInt(); // sc.nextInt();
    }
}

```

```
case 1: sav. show();  
break;  
case 2: sav. deposit();  
break;  
case 3: sav. withdraw - Bal();  
break;  
case 4: sav. check_interest();  
break;  
case 5: System.exit(0);  
break;  
default: System.out.println ("Wrong choice");
```

else

1

systems, cult. purills ("Wrong choice!"),

OUTPUT:

Case 1: Current Account

Choose Account Type (wall) | Adding two more accounts

Press (for Current Account)

Pres 8 for saving August:

c

Enter name:

Tanya

entre aéreo:

1239

Enter opening amount (must > \$00):

1000

1. Displays
 2. Deposits
 3. Withdrawals
 4. Exit

Enter your choice

2 Enter amount to deposit

1000

Name: tanya

Account-number: 12345

Amount: 1000.0

Total Amount u : 2000.0

1. Display

2. Deposit

3. Withdraw

4. Exit

Enter your choice

3 Enter amount to withdraw

1000

Name: tanya

Account-number: 12345

Amount: 2000.0

After withdrawal Balance 1000.0

1. Display

2. Deposit

3. Withdraw

4. Exit

Enter your choice

4

Case 2: savings Account-

choose account type:

Press c for current account

Press s for saving account:

s

Enter name:

Tanya

Enter accno

1234

Enter opening amount must > 500:

2000

- 2 Deposit
- 3 withdraw
- 4 Interest
- 5 Exit

Enter your choice

3

Enter amount to withdraw:

1000.0

Name: Tanya,

Acno.: 1234

Amount: 2000.0

after withdrawal, Balance: 1000.0

1. Display

2. Deposit

3. Withdraw

4. Interest

5. Exit

Enter your choice

4

Total entered amount is 1024.0

1. Display

2. Deposit

3. Withdraw

4. Interest

5. Exit

Enter your choice

5

Exit

2024
22/1/24

⑥ Package

Student.java:

```
package CIE;  
public class Student {  
    public String usn, name;  
    public int sum;  
    public Student (String usn, String name, int sum)  
    {  
        this.usn = usn;  
        this.name = name;  
        this.sum = sum;  
    }  
}
```

Internal.java:

```
package CIE;  
public class Internal extends Student {  
    public int m[] = new int [5];  
    public Internal (String usn, String name, int sum)  
    {  
        super (usn, name, sum);  
        this.m = m;  
    }  
}
```

External.java:

```
package CIE;  
import CIE.Student;  
public class External extends Student {
```

```
public int sm[] = new int [5];
public External (String usn, String name,
                  int []sm)
{
    super (usn, name, sm);
    this. sm = sm;
}
```

Main.java

```
import java.util.Scanner;
import CIE.Student;
import CIE.Internals;
import SEE.External;

public class Main {
    public static void Main (String args[])
    {
        int fm = 0;
        Scanner in = new Scanner (System.in);
        System.out.println ("Enter number of students:");
        int n = in.nextInt();
        Internals [] in = new Internals [n];
        Externals [] em = new External [n];
        Student [] stu = new Student [n];
        for (int i=0; i<n; i++)
            System.out.print ("Enter detail for student");
            (i+1) + ":";
```

~~System.out.print ("Enter name: ")~~

~~String name = in.nextLine();~~

~~System.out.print ("Enter usn: ")~~

~~String usn = in.nextLine();~~

~~System.out.print ("Enter semester: ")~~

~~int sem = in.nextInt();~~

```

int [] internal_marks = new int [5];
int [] external_marks = new int [5];
System.out.println ("Enter Marks details:");
for (int j=0; j<5; j++) {
    System.out.println ("Enter internal marks for course" + (j+1)
        + ":" );
    internal_marks[j] = in.nextInt();
}
System.out.println ("Enter external marks for course" + (j+1) +
    ":" );
external_marks[j] = in.nextInt();
}

System.out.println ();
stu[i] = new Student (usn, name, sem);
in[i] = new Internal (usn, name, sem,
    internal_marks);
em[i] = new External (usn, name, sem, external-
    marks);
}

System.out.println ("Final marks details");
for (int i=0; i<n; i++) {
    System.out.println ("Student " + (i+1) + ":" );
    System.out.println ("Name: " + stu[i].name);
    System.out.println ("USN: " + stu[i].usn);
    System.out.println ("Sem: " + stu[i].sem);
    for (int j=0; j<5; j++) {
        tm += in[i].m[j] + em[i].sm[j];
    }
    System.out.println ("Final marks of course" + (i+1) +
        ":" + tm);
}
tm = 0;
System.out.println ();

```

OUTPUT:

Enter number of student : 1

Enter details for student 1:

Enter name: Tanya

Enter USN: 12

Enter Sem: 3

Enter marks details:

Enter internal marks for course 1: 34

Enter external marks for course 1: 98

Enter internal marks for course 2: 39

Enter external marks for course 2: 97

Enter internal marks for course 3: 33

Enter external marks for course 3: 36

Enter internal marks for course 4: 33

Enter external marks for course 4: 60

Enter internal marks for course 5: 40

Enter external marks for course 5: 90

final marks details:

Student 1

Name: Tanya

USN: 12

Sem: 3

Final marks for course 1: 132

Final marks for course 2: 136

Final marks for course 3: 89

Final marks for course 4: 93

Final marks for course 5: 130

Don
30/1/24

⑦ Exception Handling:

```
import java.util.*;  
class WrongAge extends Exception {  
    public WrongAge(String str) {  
        super(str);  
    }  
}  
class Father {  
    int fage;  
    Father(int age) throws WrongAge {  
        fage = age;  
        if (fage <= 0) {  
            throw new WrongAge("age must be above 0");  
        }  
    }  
}  
class Son extends Father {  
    int sage;  
    Son(int age, int page) throws WrongAge {  
        super(page);  
        sage = age;  
        if (sage > page) {  
            throw new WrongAge("age must be less than  
            father's age");  
        }  
    }  
}
```

```
class father_son_exception {  
public static void main (String args[]) {  
    System.out.println ("Tanya D Shelly");  
    int f-age, s-age;  
    Scanner in = new Scanner (System.in);  
    System.out.println ("enter father's age");  
    f-age = in.nextInt();  
    try {  
        father f = new father (f-age);  
    } catch (WrongAge e) {  
        System.out.println ("Exception: age must be above 0");  
  
        System.out.println ("enter son's age");  
        s-age = in.nextInt();  
        try {  
            son s = new son (s-age, f-age);  
        } catch (WrongAge e) {  
            System.out.println ("Exception: " + e.getMessage());  
        }  
    }  
}
```

OUTPUT: CASE 1:

shawn 9

enter father's age

3000

-2

exception: age must be above 0

{ below 0 will not be

enter son's age

1000 below 0 will be

3

exception: age must be above 0

{ ("1000") string, too, multiple

CASE 2:

enter father's age

100

enter son's age

89

{ ("1000000000") string, too, multiple

CASE 3:

enter father's age

78

enter son's age

90

Exception: age must be less than father's age.

{ ("32") string, too, multiple

{ ("00001") string, too, multiple

{ ("1000000000") string, too, multiple

{ ("1000000000") string, too, multiple

⑧

Threads

13245

CODE:

class BMS extends Thread

public void run()

try {

for (int i=0; i<10; i++) {

System.out.println ("BMS College of Engineering");

Thread.sleep (2000);

}

}

catch (InterruptedException e)

{

System.out.println ("Exception handled");

}

}

class CSE extends Thread

public void run()

try {

for (int i=0; i<10; i++) {

System.out.println ("CSE");

Thread.sleep (10000);

}

}

catch (InterruptedException e)

{

System.out.println ("Exception handled");

}

}

```
class ThreadCollege {
```

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

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

```
    BMS b = new Thread();
```

```
    CSE c = new Thread();
```

```
    b.start();
```

```
    c.start();
```

```
    try {
```

```
        b.join();
```

```
        c.join();
```

```
}
```

```
    catch (InterruptedException e) {
```

```
        System.out.println ("Exception Handled");
```

```
}
```

```
{
```

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

```
}
```

```
5
```

```
3
```

OUTPUT:

Start

CSE

BMS College of Engineering

CSE

BMS College of Engineering 3 years, 6 semesters
CSE (Computer Science and Engineering)
CSE (Computer Science and Engineering)
CSE (Computer Science and Engineering)
CSG (Computer Science and Engineering)
CSE (Computer Science and Engineering)
CSE (Computer Science and Engineering)
CSE (Computer Science and Engineering)

Exception handled
ended

(backward slashes) allowing two methods

Dot
20/20

(backslash) following two methods

⑨ AWT (Create Label, Button and TextField in Frame AWT)

```
import java.awt.*;  
import java.awt.event.*;  
  
public class AWTExample extends WindowAdapter  
frame f;
```

```
AWTExample() {
```

```
f = new Frame();  
f.addWindowListener(this);  
Label l = new Label("Employee id:");  
Button b = new Button("submit");  
TextField t = new TextField();
```

```
l.setBounds(20, 80, 80, 30);  
l.setBounds(20, 100, 80, 30);  
b.setBounds(100, 150, 80, 30);
```

```
f.add(b);  
f.add(l);  
f.add(t);
```

```
f.setSize(400, 300);
```

```
f.setTitle("Employee info");  
f.setLayout(null);  
f.setVisible(true);
```

```
g  
public void windowClosing(WindowEvent e)  
{  
    System.exit(0);  
}
```

public static void main (String [] args) {

AWTExample awtobj = new AWTExample(),

3

2

Employee info

Employee id: 101019 Name: John Doe

+

NWT

10

Create button and add action listener for mouse click

```
import java.awt.*;  
import java.awt.event.*;  
public class EventHandling extends WindowAdapter  
implements ActionListener
```

```
Frame f;  
TextField tf;  
EventHandling() {
```

```
f = new Frame(1);  
f.addWindowListener(this);  
tf = new TextField();  
tf.setBounds(60, 50, 170, 20);  
Button b = new Button("click me");  
b.setBounds(100, 120, 80, 30);
```

```
b.addActionListener(this);
```

```
f.add(b);  
f.add(tf);  
f.setSize(300, 300);  
f.setLayout(null);  
f.setVisible(true);
```

```
public void actionPerformed(ActionEvent) {  
    tf.setText("Welcome");  
}
```

```
public void windowClosing(WindowEvent e){  
    System.exit(0);  
  
public static void main(String args[]){  
    new EventHandling();  
}  
}
```