11/12/23

OOJ LAB-1

~~Prog-1~~

#1 "Example" class To print "Hello".

CODE:

```
public class Example {
    public static void main (string args[]) {
        system.out.println ("Hello");
    }
}
```

Output: Hello

#2 To calculate sum, difference, product & quotient.

CODE:

```
public class calculations {
    void static add (int a, int b) {
        system.out.println (a+b);
    }
    void sub (int a, int b) {
        system.out.println (a-b);
    }
    void prod (int a, int b) {
        system.out.println (a*b);
    }
    void div (int a, int b) {
        system.out.println (a/b);
    }
```

```
public static void main(String args[])
{
calculations c = new calculations();
  c add (1,2);
  c sub (3,1);
  c prod (5,6);
  c div (6,2);
 }
}
```

Output : 3
2
30
3

#3 To print Fibonacci Series (n Terms)

CODE:
```
public class Fib{
    public static void main (String
                              args[]){
    int fib1 = 0;
    int fib2 = 0;
    system.out.println (fib1);
    system.out.println (fib2);
    int i, n=10;
    for (i=0; i<n-2; i++){
```

__/__/____

```
        int fib3 = fib1 + fib2;
        system.out.println(fib3);
        fib1 = fib2;
        fib2 = fib3;
    }
  }
}
```

Output:

0
1
1
2
3
5
8
13
21
34

#4 To check if a no. is prime

CODE:

```
public class prime {
  public static void main (string args[]){
    int n = 544, c = 0, i;
    for (i=1; i<n; i++){
        if (n%i == 0){
            c++;
        }
    }
}
```

__/__/____

```
if (c>=2){
system.out.println("composite");
}
else {
system.out.println("prime");
}
}
```

Output : Composite.

[another case for n=5, o/p: prime]

18/12/23

LAB-2

Prg-1

```
class grocery
{
double dal, pulses, sugar;
grocery()
{
dal = 1;
pulse = 1;
sugar = 0.5;
}
grocery (double d, double p, double s)
{
dal=d;
pulses=p;
sugar=s;
}
grocery (double one)
{
dal = one;
pulses = one;
sugar = one;
}
```

__/__/____

```
Grocery (Grocery copy)
{
dal = copy.dal;
pulses = copy.dal;
sugar = copy.dal; sugar;
}
void calc {
System.out.println("Total = "+((dal*150)
+(pulses*80)+(sugar*50)));
}
class main
{
public static void main (string a[])
grocery g1 = new Grocery();
scanner sc = new Scanner(System.in);
System.out.println("Enter the quantity of
                                    dal");
double dal = sc.next double();
System.out.println("Enter the quantity of
                                    pulses");
double pulses = sc.next double();
System.out.println("Enter the quantity
                                    of sugar");
double sugar = sc.next double();
```

__/__/__

```
grocery g2 = new grocery (dal);
grocery g3 = new grocery (dal, pulses, suga
grocery g4 = new groxery (g2);
 g1.calc();
 g2.calc();
 g3.calc();
}
```

8/1/24

LAB-3

Q.1 Quadratic

```
import java.util.Scanner;
class Quad {
  int a,b,c;
  double root1, root2,d;
  Scanner s = new Scanner (System.in);

  void input ()
{
  System.out.println ("Quadratic equation
is in the form : ax^2 + bx + c");
System.out.print ("Enter a:");
a = s.nextInt ();
System.out.print ("Enter b:");
b = s.nextInt();
System.out.print ("Enter c:");
  c = s.nextInt ();
}
void discriminant () {

    d = (b*b) - (4*a*c);
}
```

```
void calculateRoots () {
    if (d > 0)
{
 System.out.println ("Roots are real and
 unequal");
 root1 = (-b + Math.sqrt (d)) / (2*a);
 root2 = (-b - Math.sqrt (d)) / (2*a);
System.out.println ("First root is : "+root1);
System.out.println ("Second root is: "+root2);
}
else if (d == 0)
{
System.out.println("Roots are real and
                                equal");
root1 = (-b + Math.sqrt (d)) / (2*a);
System.out.println ("Root : "+root1);
}
   else
{
System.out.println ("No real solutions. Roots
are imaginary");
double real = -b / (2*a);
double imagenary = Math.sqrt(-d) / (2*a);
```

```
System.out.println("The equation has two complex roots: "+real+"+"+imag
-inary+"i and"+real+"-"+imagin
-ary+"i");
}
}
}
class Main{
 public static void (String[] args){
  Quad q= new Quad();
  q.input();
  q.discriminant();
  q.calculateRoots();
}
}
```

Output

⇒ Quadratic equation is in the form:
ax^2+bx+c
Enter a: 1
Enter b: 1
Enter c: 1
No real solution, Roots are imaginary

__/__/____

=> Quadratic eqn is in the form: ax^2+bx+c
enter a: 1
Enter b: -2
Enter c: 1
Roots are real and equal ; root: 1.0.

=> Quadratic eqn is in the form: ax^2+bx+c
Enter a: 1
Enter b: 5
Enter c: 2
Roots are real and unequal
first root is : -0.4384471871911697
Second roots is : -4.561552812808831

=> Quadratic equation is in the form: ax^2+bx+c
Enter a: -1
Enter b: -2
Enter c: -5
No real solution. Roots are imaginary
The equation has Two complex roots:
-1.0 + -2.0i and -1.0 - -2.0i

__/__/____

Q.2 Sgpa

```
import java.util.Scanner;

class Student {
    String usn;
    String name;
    int[] credits = new int[8];
    int[] marks = new int[8];

public void acceptDetails(){
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter USN:");
    usn = scanner.nextLine();

    System.out.print("Enter Name:");
    name= scanner.nextLine();

    System.out.println("Enter details for each subject:\n");
```

```
for (int i = 0; i < .credits.length ; i++){
 System.out.print ("\n Enter credits for
subject" + (i+1)+ " : " );
credits [i]= scanner.nextInt ();
System.out.print ("\n Enter marks for
Subject" + (i+1) + ":");
marks [i] = scanner.nextInt ();
}
scanner.close ();
}
public double calculate SGPA (){
 int totalCredits = 0;
 int weightedSum = 0;
 double sgpa;
 for (int i = 0; i < credits.length ; i++ {
   totalCredits += credits [i];
   int gradePoints;

gradePoints = (marks [i] /10) +1;
if (gradePoints == 11) {
      gradePoints =10;
}
else if (gradePoints <= 4) {
   gradePoints = 0;
}
```

__/__/___

```
weightedSum += gradePoints * credits[i];
}
ans = (double) weightedSum / (double)
totalCredits;
return ans;
}
}
public class sgpa{
public static void main (String[] args)
{
Scanner scanner = new Scanner(
                                System.in);
Student student = new Student();

student.acceptDetails();

System.out.println("\n Student Details");
System.out.println("USN:" + student.usn);
System.out.println("Name:" + student.name);

double sgpa = student.calculateSGPA();
System.out.println("\n SGPA:" + sgpa);
scanner.close();
}
}
```

__/__/____

Output

Enter USN: 1bm22cs008
Enter Name: dhanush k

Enter details for each subject:

Enter credits for subject 1: 4
Enter marks for subject 1: 79

Enter credits for subject 2: 4
Enter marks " " 2: 90

Enter credits for subject 3: 3
" marks " " 3: 89

Enter credits for subject 4: 3
" marks " " : 88

Enter credits for subject 5: 2
" marks " " 5: 98

Enter more 1 6: 1
6: 99

__/__/____

Enter credits for subject 7 : 1
" marks " " 7 : 99

Enter credits for subject 8 : 4
" marks " " 8 : 89

Shri 08/01/24

A#

__/__/__

Book

Q.3

```
import java.util.Scanner;
class Book {
  String name;
  String author;
  double price;
  int num_pages;

Book (String name, String author, double price,
int num_pages) {
  this.name = name;
  this.author = author;
  this.price = price;
  this.num_pages = num_pages;
}

void setDetails () {
  name = name;
  author = author;
  price = price;
  num_pages = num_pages;
}
```

__/__/____

```
void getDetails (){

Scanner S = new Scanner (System.in);
System.out.print ("Enter Book Name :");
 name = S.nextLine ();
System.out.print ("Enter Author Name: ");
author = S.nextline ();
System.out.print ("Enter Price :");
price = S.nextDouble ();
System.out.print ("Enter Number of Pages :");
 num_pages = S.next Int ();
System.out.println (" - - - - - - - ");
}
public String toString (){
  return ("Book Name :" + name + "\n
Author Name : "+author+"\n Price :"+price"
\n Number of pages :"+num_pages);
 }
}

class Book Main {
public static void main (String args[])
{
Scanner S= new Scanner (system.in);
int n, i;
```

```
System.out.println("Enter Number of Books");
m=s.nextInt();
Book[] books = new Book[m];
for (i=0; i<m; i++) {
System.out.println("Enter details of book"
+(i+1));
books[i]=new Book(" "," ",0.0,0);
books[i].getDetails();
}
for (i=0; i<m ; i++){
System.out.println("Details of book"+(i+1));
System.out.println(books[i]);
}
}
}
```

Output.

Enter number of Books
2
Enter details of book 1.
Enter book name: it ends with us
Enter author: colleen Hoover
Enter price : 250
Enter number of pages : 200

__/__/____

Enter details of book :
Enter book name : Tuesday love
Enter author : Ana Huang
Enter price : 380
Enter number of pages : 400

LAB-4

22/1/24

Q LAB Program - 4

```
abstract class Shape {
    // Two integers representing dimensions
    protected int dimension1;
    protected int dimension2;

    // Constructor
    public Shape(int dimension1, int
    dimension 2) {
        this.dimension1 = dimension1;
        this.dimension2 = dimension 2;
    }

    public abstract void printArea();
}
class Rectangle extends Shape {

    public Rectangle(int length, int width) {
        super(length, width);
    }
```

__/__/____

```
    public void printArea(){
    int area = dimension1 * dimension2;
    System.out.println("Area of Rectangle:"
+ area);
    }
}
class Triangle extends Shape{

    public Triangle(int Base, int height){
    super(base, height);
    }

    public void printArea(){
    double area = 0.5 * dimension1 * dimension2;
    System.out.println("Area of Triangle:" +
area);
    }
}
class Circle extends Shape{

    public Circle(int radius){
    super(radius, 0);
    }
```

__/__/____

```
public void printArea(){
double area = Math.PI * dimension1 *
dimension1;
System.out.println("Area of Circle:" +
area);
}
}
public class Main{
public static void main(String[] args){
  Rectangle rectangle = new Rectangle(5,10);
  Triangle triangle = new Triangle(8,6);
  Circle circle = new Circle(4);

  rectangle.printArea();
  triangle.printArea();
  circle.printArea();
}
}
```

__/__/____

Output

Area of rectangle : 50
" " triangle : 24.0
circle : 50.26548245743669

__/__/____

LAB - Program 5

```
import java.util.Scanner;

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

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

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

__/__/____

```
public void displayBalance (){
System.out.println ("Balance for account"
+accountNumber + ": " + balance );
}

public void depositInterest (double rate){
 if ("Savings".equals (accountType)){
  double interest = balance * (rate / 100);
  balance += interest;
  System.out.println ("Interest deposited.
Updated balance: " + balance);
} else {
 System.out.println ("No interest for
current account.");
 }
}

public void withdraw (double amount)
{
 if (amount <= balance){
   balance -= amount;
 System.out.println ("Withdrawal
 successful. Updated balance: " + balance);
} else {
```

```
System.out.println("Insufficient funds for
withdrawal.");
    }
  }
}

class SavAcct extends Account {
  public SavAcct (String customerName,
  long accountNumber, double balance){
   super(customerName, accountNumber,
"Savings", balance);
  }
}
class CurrAct extends Account {
  double minBalance;
  double serviceCharge;

  public CurrAct (String customerName,
  long accountNumber, double balance,
double minBalance, double serviceCharge)
{
super (customer Name , accountNumber,
"Current", balance);
  this.minBalance = minBalance;
  this.serviceCharge = serviceCharge;
}
```

__/__/____

```
@Override
public void withdraw(double amount) {
 if (amount <= balance - minBalance){
    balance -= amount;
  System.out.println("Withdrawal
successful. Updated balance: "+balance);
} else {
System.out.println("Insufficient funds
for withdrawal. Minimum balance
requirement not met.");
 }
}

min public void checkMinimumBalance()
{
 if (balance < minBalance){
    balance -= serviceCharge;
System.out.println("Service charge
imposed for falling below minimum
balance. Updated balance: "+balance);
   }
}
}
```

__/__/__

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

 SavAcct savingsAccount = new SavAcct ("John
 Doe", 123456789, 1000.0);

 CurrAcct currentAccount = new CurrAcct
 ("Jane Doe", 987654321, 2000.0, 500.0,
 50.0);

 System.out.println("Savings Account
 Operations:");
  savingsAccount.displayBalance ();
  savingsAccount.deposit (500.0);
  savingsAccount.depositInterest (5.0);
  savingsAccount.withdraw (200.0);

  System.out.println("\n Current Account
  Operations:");
  currentAccount.displayBalance ();
  currentAccount.deposit (1000.0);
  currentAccount.withdraw (800.0);
  currentAccount.checkMinimumBalance ();
   scanner.close ();
}
}
```

__/__/___

Output:

Savings account operations:
Balance for amount 12345678 9: 1000
deposit successful. updated balance: 1500
internal deposit, updated balance: 1575.0
withdrawal successful. update balance:
1375.0

LAB-5

29/ 1/ 24

Package

```
Package CIE;
public class student {
    string USN;
    string name;
    int sem;
    public student (string USN, string name,
    int sem);
    this.USN = USN;
    this.name = name;
    this.sem = sem;
    }
}
```

```
Package CIE;
public class internals extends student{
    public int [] internal marks;
    public internals (string USN, string
    name, int sem, int [] internal
    marks) {
    super.(USN, name, sem);
    this.internal marks = internal marks;
    }
}
```

__/__/__

```
Package SEE;
import CIE.student;
public class external extends student{
  public int [] see Marks;
  public External (String USN, string
  name, int sem, int [].see marks){
    super.(USN, name, sem)
    this.see Marks = see marks;
  }
}
import CIE.Internals;
import SEE.~~int~~ Externals;
import Java.util.Scanner;
public class final marks {
  public static void main [String [] args){
   Scanner.scanner = new Scanner (syste
   System.out.println("Enter no. of students");
   int n = Scanner nextInts;
   internals [] cie students = new Internal
   external [] see students = new external [n];
   for (int i = 0; i<n; i++) {
    SOP ("enter CIE details "+(i+1));
    SOP ("USN:");
    ~~SOP~~ String USN = scanner.next ();
```

```
SOP("Name:");
String name = scanner.next;
SOP("sem:");
int sem = scanner.next;
int [] cieMarks = new int[5];
SOP("enter cie marks for 5 courses");
for(int j=0; j<5; j++){
 cie marks[j] = scanner.next.int(5);
}
cie students[i] = new internals
(USN, name, cie); }
for int }
 for(int i=0; i<n; i++){
  SOP("Enter SEE details" + (i+1));
  SOP("USN");
 String USN = scanner.next();
SOP("Name:");
String name = Scanner.next()
 SOP("sem:");
 Int Sem = scanner.next int();
int[] SEE marks. new int[5];
SOP("enter SEE marks for semester");
for(int j=0; j<5; j++){
 see marks[j] = scanner next int();
}
```

```
see student [i] = new external (usn,
name, sem, see marks)
}
SOP("\n find marks of students:");
for (int i=0; i<n; i++)
SOP("\n details of student"+(i+1));
SOP("USN, "+" estudent[i].usn);
SOP("name"+"estudent[i].name);
SOP("sem", "+"cie students(i)].sem);
SOP("CIE marks:")
for (int j=0; i<8; j+1){
SOP( cie students(i). internal marks(j)");
}
SOP("\n SEE marks:");
for (int j=0; j<5; j++){
SOP(see students[i] see marks[j]+" ");
}
}
```

__/__/__

Output

Enter number of students : 1
Enter details for CIE of student 1
USN : 1BM22CS081
Name : darshna
Sem : 2
Enter CIE marks for 5 courses :
40
38
40
40
35
Enter details of SEE of student 1.
USN : 1BM22CS081
Name : darshna
Sem : 4
Enter see marks for 5 courses :
89
78
65
69
95

__/__/____

Final marks of students:
details of student 1
USN : 1BM22CS081
CIE marks:
40 38 40 40 35
SEE marks:
89 78 65 69 95

16.02.24

19/2/24

LAB=6

Q Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every 10 seconds and another displaying "CSE once every 2 seconds

```
class BMS implements Runnable{
  public void run (){
  while (true) {
    Try {
    system.out.println ("BMS college of Engg");
     Thread.sleep(10000);
    } catch (InterruptedException e) {
      e.printStackTrace();
    }
  }
 }
}
class CSE implements Runnable {
    public void run() {
     while (true) {
     Try {
     system.out.println("CSE");
     Thread.sleep(2000);
  } catch (InterruptedException e) {
```

__/__/____

```
            e.printStackTrace();
        }
    }
}
}
}
public class Main {
    public static void main(String[] args){
    Thread t1 = new Thread(new BMS College
                of Engineering());
    Thread t2 = new Thread(new CSE());
    t1.start();
    t2.start();
}
}
```

O/P:

BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE

__/__/__

CSE
BMS College of Engineering.
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering

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

```
import java.util.Scanner;
class WrongAge extends Exception.
{
 public WrongAge (String message)
 {
   super (message);
 }
}
class Father
{
 int fatherAge;
Father() throws WrongAge
{
 Scanner s = new Scanner (System.in);
System.out.println ("Enter Father's Age:");
fatherAge = s.nextInt();
if (FatherAge < 0)
{
 throw new WrongAge ("Age cannot be
 negative");
}
}
```

__/__/____

```
void display()
{
System.out.println("Father Age is : " +
fatherAge);
}
}

class Son extends Father
{
 int SonAge;
 Son() throws WrongAge
 {
  super();
 Scanner s = new Scanner(System.in);
 System.out.println("Enter Son's age:");
 SonAge = s.nextInt();
 if (SonAge > FatherAge)
{
 throw new WrongAge("Son's age cannot
be greater than Father's age");
}
else if (SonAge == FatherAge)
{
throw new WrongAge("Son's age cannot be
equal to Father's age");
}
```

__/__/____

```
else if (sonAge < 0)
{
throw new WrongAge("Age cannot be
negative");
}
void display()
{
 super.display()
 System.out.println("Son's age is:"+
sonAge);
}
}
public class main
{
public static void main (String[] args)
 {
  try {
  Son s= new Son();
   s.display();
 }
catch (WrongAge e)
 {
System.out.println(e.getMessage());
 }
}
}
```

__/__/____

Output :

Enter Father's age : 40
Enter Son's age : 18
Father's age is : 40
Son's age is : 18

Enter Father's age : 30
Enter Son's age : 30
Son's age cannot be equal to Father's age.

Enter Father's age : -20
Age cannot be negative

19.02.24

26/2/24

LAB=7

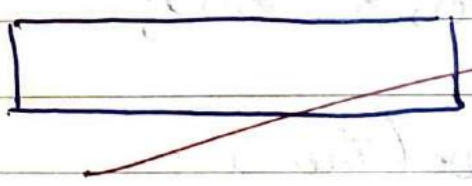
Q.1 Create label, button and TextField in a Frame using AWT.

```
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);
t.setBounds (20, 100, 80, 30);
b.setBounds (100, 100, 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);
}
public void windowClosing(WindowEvent e){
    System.exit(0);
}

public static void main(String[] args){

AWTExample awt_obj = new AWTExample();
}
}
```

Output

Employee Id :



__/__/____

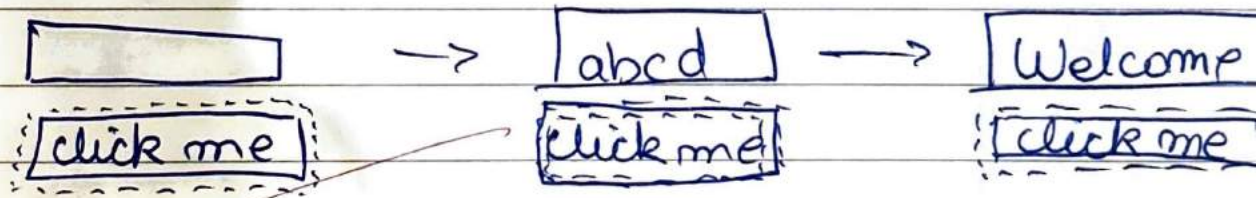
Q.2 Create a button and add a action listener for Mouse click.

```
import java.awt.*;
import java.awt.event.*;
public class EventHandling extends
Window Adapter implements ActionListener
{
frame f;
TextField tf;
EventHandling (){
    f = new Frame();

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

Output



26.02.24