

A decorative banner featuring five large, bold, black letters ('I', 'N', 'D', 'E', 'X') arranged horizontally. Each letter is enclosed in a white square frame with a black border, and the letters are slightly overlapping each other.

NAME: Dipesh Sah SEM: 3 SEC.: B ROLL NO.: 092 SUB.: OOT Kab.

Lab - 1

Prog. 1) Basic - Printing - 'Hello World'.

⇒ class Hello

{ public static void main (String args[])

{ System.out.println("Hello World");

}

⇒ Output :→ Hello World.

Prog. 2) Checking if a number is PRIME or not.

Syntax ⇒ class Prime

{

 public static void main (String args[])

 int n = 17;

 int count = 0;

 for (int i = 1, i <= n; i++)

 if (n % i == 0) {

 count++;

}

}

```
if (count == 2)
```

{

else

{

{

{

{

```
    system.out.println("It is a  
    Prime Number.");
```

```
else
```

```
    system.out.println("It is  
    not a Prime Number.");
```

Output: It is a Prime Number.

Prog. 3) Fibonacci Series.

⇒ Syntax :-

```
class Fib {
```

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

```
        int n1 = 0, n2 = 1, n3, i, count = 10;
```

```
        System.out.println(n1 + " " + n2);
```

```
        for (i = 2; i < count; i++)
```

```
            n3 = n1 + n2;
```

```
            System.out.println(" " + n3);
```

```
            n1 = n2;
```

```
            n2 = n3;
```

```
}
```

⇒ Output :-

0 1 1 2 3 5 8 13 21 34

Prog. 4) Calculation of Numbers.

→ Syntax :-

```
class Calci  
{
```

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

```
        int a = 10, b = 20;
```

```
        int add = a + b;
```

```
        int sub = b - a;
```

```
        int div = b / a;
```

```
        int prod = b * a;
```

```
        System.out.println ("Sum = " + add);
```

```
        System.out.println ("Sub = " + sub);
```

```
        System.out.println ("Div = " + div);
```

```
        System.out.println ("Prod = " + prod);
```

```
}
```

```
}
```

⇒ Output :-

Sum = 30

Sub = 10

Div = 2

Prod = 200

SD

8/11/2024

Write a Java program to create a class student
Prog. 1) with USN, name, marks (6 subjects). Include methods
to accept student details & display appropriate details
Syntax: import java.util.Scanner;

```
class Student {
```

```
    String USN;
```

```
    String name;
```

```
    int [] marks = new int [6];
```

```
    void acceptDetails () {
```

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

```
        System.out.print("Enter USN:");
```

```
        USN = scanner.next();
```

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

```
        name = scanner.next();
```

```
        System.out.println("Enter marks  
for all 6 subjects:");
```

```
        for (int i=0; i<6; i++) {
```

```
            System.out.print("Subject" +  
(i+1) + ":");
```

```
            marks[i] = scanner.nextInt();
```

```
}
```

3

```
double calculatePercentage () {
```

```
    int totalMarks = 0;
```

```
    for (int marks : marks) {
```

```
        totalMarks += mark;
```

```
}
```

```
    return (double) totalMarks / 6;
```

```
}
```

```
void displayDetails () {
```

```
    System.out.println ("Student details:");
```

```
    System.out.println ("USN: " + USN);
```

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

```
    System.out.println ("Percentage: " +  
        calculatePercentage () + "%");
```

```
}
```

```
}
```

~~```
public class Student_Det {
```~~~~```
    public static void main (String [] args) {
```~~~~```
 Scanner scanner = new Scanner (System
 .in);
```~~~~```
        System.out.print ("Enter the number  
            of students: ");
```~~~~```
 int numStudents = scanner.nextInt();
```~~~~```
        Student [] students = new Student  
            [numStudents];
```~~

```
for(int i = 0 ; i < numStudents ; i++) {  
    students[i] = new Student();  
    System.out.println("In Enter the  
details for student"+(i+1)+":");  
    students[i].acceptDetails();  
}  
  
System.out.println("In Details of all  
students :");  
for (Student student : students) {  
    student.displayDetails();  
    System.out.println();  
}  
}
```

Output: Enter the number of students : 2

Enter the details for student 1 :-

Enter USN : 1BM22CS092

Enter Name : Dinesh

Enter the marks for 6 subjects :-

Subject 1 : 78

Subject 2 : 85

Subject 3 : 96

Subject 4 : 79

Subject 5 : 89

Subject 6 : 84

Enter details for student 2 :

Enter USN : 1BM22CS109

Enter Name : Marsh

Enter marks for 6 subjects :

Subject 1 : 98

Subject 2 : 96

Subject 3 : 94

Subject 4 : 93

Subject 5 : 99

Subject 6 : 92

Details of all Students :

Student details 1 :

USN : 1BM22CS092

Name : Dipesh

Percentage : 86.83

Student details 2 :

USN : 1BM22CS109

Name : Marsh

Percentage : 95.33

Ques. 2) Create a class Book that 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.

Syntax: import java.util.Scanner;

class Books {

Scanner s = new Scanner(System.in);
String name;
String author;
int price;
int numPages;

~~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 bookName = "Book Name :" +  
        this.name + "\n";  
    String authorName = "Author Name :" +  
        this.author + "\n";  
    String bookPrice = "Price :" + this.  
        price + "\n";  
    String pages = "Number of Pages :" +  
        this.numPages + "\n";  
    return bookName + authorName +  
        bookPrice + pages;  
}
```

```
}  
  
class BookProgram {  
    public static void main (String args[]) {  
        Scanner s = new Scanner (System.in);  
        int n;  
        System.out.println ("Enter the  
            number of books : ");  
        n = s.nextInt ();  
        Books b[] = new Books [n];  
        for (int i=0; i < n; i++) {  
            System.out.println ("\nDetails of  
                Book - " + (i+1) + ":");  
            System.out.println ("Enter the name
```

of Book :- ");

String name = s.next();

System.out.println("Enter the name
of Author :- ");

String author = s.next();

System.out.println("Enter the price :-");
int price = s.nextInt();

System.out.println("Enter the number
of Pages :- ");

int numPages = s.nextInt();

b[i] = new Books(name, author,
price, numPages);

{

// Displaying book details

System.out.println("In Details of all
books : ");

for (Books book : b) {

System.out.println(book.toString());

{

}

Output: Enter the number of books : 2

Details of Book_1 :

Enter the name of Book : Discrete Mathematics

Enter the name of Author :- Chaitra

Enter the Price :- 345

Enter the number of Pages :- 428

Details of Book 2 :

Enter the name of Book :- Java-Programming

Enter the name of Author :- Shravya

Enter the price :- 420

Enter the number of Pages :- 386 .

Details of all books :

Book Name :- Discrete Mathematics .

Author Name : Chaitra

Price : 345

Number of Pages : 428

Book Name : Java-Programming

Author Name : Shravya

Price : 420

Number of Pages : 386

Ques.3) Write a Java program to execute the quadratic equation roots.

```
Syntax: import java.lang.Math;  
import java.util.Scanner;  
class Quadratic {  
    Scanner s = new Scanner(System.in)  
    System.out.println("Enter equation of  
form ax2+bx+c");  
    this.a = s.nextInt();  
    this.b = s.nextInt();  
    this.c = s.nextInt();  
}  
void calculateRoot() {  
    int d = b*b - 4*a*c  
    if (d == 0) {  
        double root1 = -b/(a*c);  
        System.out.println("Roots are " +  
root1 + " and " + root2);  
    }  
    else if (d > 0) {  
        double root1 = (-b + Math.sqrt(d)) /  
(2*a);  
        double root2 = (-b - Math.sqrt(d)) /  
(2*a);  
    }  
}
```

```
        system.out.println("R1 = " + r1);
        system.out.println("R2 = " + r2);
    }
    else if (d == 0) {
        system.out.print("The equation
        has real and equal solutions");
        double r1 = -b/(2*a);
        double r2 = -b/(2*a);
        system.out.print("R1 = R2 = " + r1);
    }
    else if (d < 0) {
        system.out.print("The equation
        has unreal solutions.");
    }
}
```

⇒ Output ⇒

Enter the values of a, b, c respectively:

2, 4, 5

- The equation has unreal solutions.

Enter the values of a, b, c respectively:

1 -7 10

- The equation has two real and different solutions.

$$R_1 = 5.0$$

$$R_2 = 2.0$$

- Enter the values of a, b, c respectively :

~~1 -4 4~~

The equation has real and equal solution

$$R_1 = R_2 = 2.0$$

Prog. 1) 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 contains the method printArea() that prints the area of the given shape.

Syntax: abstract class Shape {

 private int length;

 private int width;

 public Shape(int length, int width) {

 this.length = length;

 this.width = width;

 }

 public abstract void printArea();

}

class Rectangle extends Shape {

 public Rectangle(int length, int width) {

 super(length, width);

}

@Override

```
public void printArea() {  
    int area = length * width;  
    System.out.println("Area of  
    Rectangle : " + area);  
}
```

3

class Triangle extends Shape {

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

@Override

```
public void printArea() {  
    double area = 0.5 * length * width;  
    System.out.println ("Area of Triangle  
    " + area);  
}
```

3

class Circle extends Shape {

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

@Override

```
public void printArea() {  
    double area = Math.PI * length * length;  
}
```

```
system.out.println("Area of circle:" + area);  
}  
}  
  
public class Main {  
    public static void main (String [] args) {  
        Rectangle rectangle = new Rectangle(4,  
                                             rectangle.printArea());  
  
        Triangle triangle = new Triangle (3, 6);  
        triangle.printArea();  
  
        Circle circle = new Circle (7);  
        circle.printArea();  
    }  
}
```

Output: ~~Area of Rectangle: 20.0
Area of Triangle: 9.0
Area of Circle: 153.98~~

Prog.2) 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.

Syntax:- import java.util.Scanner;

class Account {

 String customerName;

 long accountNumber;

 String accountType;

 double balance;

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

}

 public void displayBalance () {

 System.out.println ("Account Balance : Rs " + balance);

}

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

```
class SavAcct extends Account {  
    double interestRate;  
    public SavAcct (String customerName,  
    long accountNumber, double balance,  
    double interestRate) {  
        super (customerName, accountNumber,  
        savings, balance);  
        this.interestRate = interestRate;  
    }  
}
```

```
public void computeInterest () {  
    double interest = balance * interestRate  
    / 100; }
```

deposit (interest);
system.out.println ("Interest Rs" +
"interest" + "computed & deposited");

{

public void withdraw (double amount) {

if (balance >= amount) {

balance -= amount;

system.out.println ("Withdrawal
of Rs" + amount + "successful");

{

else {

system.out.println ("Insufficient
funds. Withdrawal failed.");

{

{

{

class current extends Account {

double minBalance;

double serviceCharge;

public current (String customerName,
long accountNumber, double balance,
double minBalance, double serviceCharge)

super (customerName, accountNumber,
currentBalance);

this.minBalance = minBalance;

this.serviceCharge = serviceCharge;

{

public void withdraw (double amount) {

if (balance - amount >= minBalance) {

super.withdraw (amount);

{

else {

System.out.println ("Withdrawal
failed. Minimum Balance not
maintained. Service charge of
Rs" + serviceCharge + "imposed");
balance -= serviceCharge;

{

{

{

public class Bank {

public static void main (String [] args) {

Scanner scanner = new Scanner (sys-
tem.in);

SavAct savingsAccount = new SavAct (
"John", 123456789, 1000.00, 5.0);

savingsAccount.displayBalance ();

savingsAccount.deposit (5000);

savingsAccount.computeInterest ();

. savingsAccount.displayBalance();

current currentAccount = new currentAccount("John", 248678924, 1500.0, 1000.0, 20.0);

currentAccount.displayBalance();

currentAccount.deposit(300.0);

currentAccount.withdraw(700.0);

currentAccount.displayBalance();

scanner.close();

}

}

Output: Account Balance : Rs 1500.0

Deposit of Rs 500.00 successful.

Deposit of Rs 75.00 successful.

Interest of Rs 175.00 computed and deposited.

Withdrawal of Rs 200.00 successful.

Account Balance : Rs 1375.00

Account Balance : 1500.00

Deposit of Rs 300.00 successful.

Withdrawal of Rs 700.00 successful.

Account Balance : Rs 1100.00.

Q1
S1/2
2

⇒ Syntax for Main.java outside the subfolders.

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

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

```
        System.out.print("Enter the number  
        of students : ");  
        int n = scanner.nextInt();  
        scanner.nextLine();
```

```
        Internals [] internaData = new Internals[n];  
        External [] externalData = new External[n];
```

```
// Initialize data for students  
for (int i=0; i<n; i++) {  
    //System.out.print("Enter USN:");  
    //String  
    System.out.println("In Enter the  
    details for Student"+(i+1)+":");  
    System.out.print("Enter USN:");  
    System.out.print
```

```
String user = scanner.nextLine();
```

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

```
System.out.print("Enter Semester :");  
int sem = scanner.nextInt();
```

```
System.out.println("Enter Internal  
Marks for 5 subjects :");
```

```
int[] internalMarks = new int[5];
```

```
for (int j = 0; j < 5; j++) {  
    System.out.print("Subject" +  
        (j + 1) + ": ");
```

```
internalMarks[j] = scanner.  
nextInt();
```

```
}
```

```
System.out.println("Enter External  
Marks for 5 subjects :");
```

```
int[] externalMarks = new int[5];
```

```
for (int j = 0; j < 5; j++) {
```

```
System.out.print("Subject" +  
    (j + 1) + ": ");
```

```
externalMarks[j] = scanner.nextInt();
```

```
}
```

// Create Internals and External objects for the student

internalsData[i] = new Internals(
 usn, name, sem, internalMarks);

externalData[i] = new External(
 usn, name, sem, externalMarks);

}

// Display final marks of students

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

 System.out.println("In Details of
 Student " + (i+1) + ":");

 System.out.println("USN: " + internals
 Data[i].usn);

 System.out.println("Name: " +
 internalsData[i].name);

 System.out.println("Semester: " +
 internalsData[i].sem);

 System.out.println("Internal Marks:
 " + arrayToString(internalsData[i].
 internalMarks));

 System.out.println("External Marks:
 " + arrayToString(externalsData[i].
 externalMarks));

 System.out.println("Total Marks
(IE+SSE): " + calculateTotalMarks)

```
{ internalData[i].internalMarks,  
  externalData[i].externalMarks));  
}  
scanner.close();  
}  
  
// Utility method to calculate total marks  
private static int calculateTotalMarks(  
    int[] internalMarks, int[] externalMarks) {  
    int total = 0;  
    for (int i=0; i < internalMarks.length;  
        i++) {  
        total += internalMarks[i] + externalMarks[i];  
    }  
    return total;  
}  
  
// Utility method to convert an array to  
// string for printing.  
private static String arrayToString(int[]  
    array) {  
    StringBuilder sb = new StringBuilder();  
    sb.append('[');  
    for (int i=0; i < array.length; i++) {  
        sb.append(array[i]);  
        if (i != array.length - 1) {  
            sb.append(", ");  
        }  
    }  
    sb.append(']');  
    return sb.toString();  
}
```

```
        sb.append(str: " , ");  
    }
```

```
    }  
    sb.append(str: "]");  
    return sb.toString();
```

{

{

Syntax for Internals.java inside the
subfolder named CIE.

```
package CIE;
```

```
public class Internals extends Student {  
    public int [] internalMarks;
```

```
    public Internals (String usn, String name,  
                     int sem, int [] internalMarks) {  
        super (usn, name, sem);  
        this. internalMarks = internalMarks;
```

{

{

Syntax for Student.java inside the sub-folder named CIE.

package CIE;

public class Student {

 public String usn;

 public String name;

 public int sem;

 public Student (String usn, String name,
 int sem) {

 this.usn = usn;

 this.name = name;

 this.sem = sem;

}

}

Syntax for External.java inside the SE subfolder named SEE.

package SEE;

import CIE.Student;

public class External extends Student {

public int [] externalMarks;

public External (String usn, String name,
int sem, int [] externalMarks) {
super (usn, name, sem);
this . externalMarks = externalMarks;

}

3

Output :> Enter the number of students : 2

Enter the details for Student 1 :

Enter USN : 1BM22CS092

Enter Name : Dipesh Sah

Enter Semester : 3

Enter Internal Marks for 5 subjects :

Subject 1 : 94

Subject 2 : 97

Subject 3 : 92

Subject 4 : 89

Subject 5 : 93

Enter External Marks for 5 subjects :

Subject 1 : 87

Subject 2: 91

Subject 3: 89

Subject 4: 94

Subject 5: 97

- Enter the details for student 2

Enter USN: 1BM22CS104

Enter Name: Harsh

Enter Semester: 3

Enter Internal Marks for 5 subjects:

Subject 1: 97

Subject 2: 94

Subject 3: 96

Subject 4: 91

Subject 5: 89

Enter External Marks for 5 subjects:

Subject 1: 95

Subject 2: 92

Subject 3: 89

Subject 4: 84

Subject 5: 87

Details of Student 1:

USN: 1BM22CS092

Name: Dipesh Sah

Semester: 3

Internal Marks: [94, 97, 92, 89, 93]

External Marks: [87, 91, 89, 94, 97]

Total Marks (CIET+SEE): 923

Details of Student 2:

USN: 1BM22CS092

Name: Harsh

Semester: 3

Internal Marks: [97, 94, 96, 91, 89]

External Marks: [95, 92, 89, 84, 87]

Total Marks (CIET+SEE): 914

Ques. 1) 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.

Syntax: class WrongAge1 extends Exception {
 WrongAge1 (String message) {
 super (message);
 }
}

~~class WrongAge2 extends Exception {
 WrongAge2 (String message) {
 super (message);
 }
}~~

3

```
class Father {  
    int bapuage;
```

```
    Father (int bapuage) {  
        this.bapuage = bapuage;  
    }
```

{

```
class Son extends Father {  
    int betaaage;  
    Father father;
```

```
    Son (int bapuage, int betaaage) throws  
        WrongAge2, WrongAge2 {  
        super (bapuage);  
        this.betaaage = betaaage;  
        if (this.bapuage <= 0) {  
            throw new WrongAge1 ("Father  
not born ??");  
        }
```

{

```
    else if (this.bapuage <= this.betaaage)  
        throw new WrongAge2 ("Son  
is father or father is son");  
    } else {  
    }
```

{

3

```
class Main {  
    public static void main (String args[]) {  
        try {  
            Son s = new Son(84, 5);  
        } catch (WrongAge1 e) {  
            System.out.println(e);  
        } catch (WrongAge2 e) {  
            System.out.println(e);  
        }  
        try {  
            Son s = new Son(-1, 5);  
        } catch (WrongAge1 e) {  
            System.out.println(e);  
        } catch (WrongAge2 e) {  
            System.out.println(e);  
        }  
    }  
}  
  
Output:→ WrongAge2 : Wrong Son Age  
WrongAge1 : Wrong Father Age.
```

Prog. 2.) Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "ESE" once every two seconds.

Syntax is class B MS extends Thread {

```
public void run() {
    while (true) {
        try {
            System.out.println("BMS College
of Engineering.");
            Thread.sleep(10000);
        } catch (InterruptedException e) {
        }
    }
}
```

```
        catch (InterruptedException e) {  
            System.out.println(e);  
        }  
    }  
}
```

3

2

2

~~class CSE extends Thread~~

```
public void run() {
```

while (true);

try {

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

```
Thread.sleep(2000);
```

{

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

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

}

}

}

```
class Main {
```

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

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

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

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

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

}

}

Output: BMS College of Engineering.

CSE

CSE

~~CSE~~

~~CSE~~

Dr. M. 19/10/24

BMS College of Engineering.

Prog.1) Create a Label, button and Textfield a frame using AWT.

Syntax: 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);

l.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();

}

3

Output:

Emp.3024

Submit

→ Button

↓
Text Field

Prog-2) Create a button and add a action listener for mouse click.

Syntax:

```
import java.awt.*;  
import java.awt.event.*;  
public class EventHandling extends  
WindowAdapter implements ActionListener  
Frame f;  
TextField tf;  
EventHandling () {
```

// create components

```
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);
```

// register listener

```
b.addActionListener (this);  
tf.setBounds (60, 50, 170, 20);  
Button b = new Button ("click me");
```

If add components and set size, layout and visibility

```
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);
```

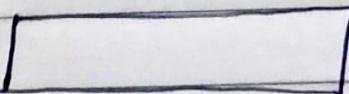
```
}
```

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

```
    new EventHandling();
```

```
}
```

```
(3)
```



Click me

SSD

29/3/2021

Output: