

① Hello world

```
class Main {
    public static void main(String[] args) {
        System.out.println("Hello world");
    }
}
```

Output: Hello world

② Class Fibo // Fibonacci

```
public static void main(String[] args) {
    int n1=0, n2=1, n3, i;
    System.out.print(n1 + " " + n2);
    for (i=2; i<10; ++i) {
        n3 = n1+n2;
        System.out.print(" " + n3 + " ");
        n1 = n2;
        n2 = n3;
    }
}
```

③ The value of 5th term is 3

Arithmetic Operations

```
class Math {
    public static void main(String[] args) {
        System.out.print("Enter the num1: ");
        double num1 = 10.5;
        double num2 = 84.8;
        double add = num1 + num2;
        double sub = num1 - num2;
        double mult = num1 * num2;
        double div = num1 / num2;
    }
}
```

```
System.out.println("add "+add);  
System.out.println("sub "+sub);  
System.out.println("mult "+mult);  
System.out.println("div "+div);
```

3 add output = add 44.89 / sub 24.89 / MUL 366.845
div 3:32

Prime numbers

Class Fibos

public static

Class' Prime &

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

8

int i,j,m=0,flog=0;

int m=10;

$$\text{int } m = m/2;$$

if ($m == 0$ || $m == 1$)

8

```
System.out.println("not prime");
```

۳

else

9

~~for (i=2; i<=m; i++)~~

~~If $(m \cdot \% \cdot i = 0)$~~

```
System.out.print("not prime");
```

$$\text{flag} = 1;$$

breakfast

3

٢

14

$$\{ \quad j = 0 \}$$

26

3

1

Output

* (the value of 5th term is 3) * the number is not prime

(3) Constructor type

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

```
class Grocery {
```

```
    double dal, pulses, sugar;
```

```
Grocery () {
```

```
    dal = 1;
```

```
    pulses = 1.5;
```

```
    sugar = 0.5;
```

```
}
```

```
Grocery (double a) {
```

```
    dal = a;
```

```
    pulses = a;
```

```
    sugar = a;
```

```
Grocery (double a, double b, double c)
```

```
{
```

```
    dal = a;
```

```
    pulses = b;
```

```
    sugar = c;
```

```
}
```

```
Grocery (Grocery obj)
```

```
{
```

```
    dal = obj.dal;
```

```
    pulses = obj.sugar;
```

```
}
```

```
double sale () {
```

return dal * 10 + pulses * 50 + sugar * 25;

```
}
```

class main {

 public static void main(String args[]) {

 System.out.println("Enter the quantity of all items");

 Scanner s = new Scanner(System.in);

 double a = s.nextDouble();

 Grocery g1 = new Grocery();

 Grocery g2 = new Grocery(a);

 Grocery g3 = new Grocery(a, 2, 2.5);

 Grocery g4 = new Grocery(g3);

 System.out.println(g2.calcl());

 System.out.println(g2.calcl());

 System.out.println(g3.calcl());

 System.out.println(g4.calcl());

}

Output

2 2 2

10.5

4.20

4.20

Create a class Book that contains name, author, price and num-pages. include a constructor to set values. include String methods to display details of book.

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

```
class Book {
```

```
    String name;
```

```
    String author;
```

```
    double price;
```

```
    int num_page;
```

```
    Book(String name, String author, double price, int num-page)
```

```
        -g e) {
```

```
            this.name = name;
```

```
            this.author = author;
```

```
            this.price = price;
```

```
            this.num_page = num_page;
```

```
}
```

```
    public String toString() {
```

```
        String Author, name, price, num_page;
```

```
        name = "Book name: " + this.name + "\n";
```

```
        Author = "Author name: " + this.author + "\n";
```

```
        price = "price: " + this.price + "\n";
```

```
        num_page = "Num of pages: " + this.num_page + "\n";
```

```
        return name + Author + price + num_page;
```

```
}
```

```
class Main {
```

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

~~Scanner scanner = new Scanner(System.in);~~~~int n;~~~~String name;~~~~String author;~~~~double price;~~~~int num_page;~~

② Java
marks
import
class

```
System.out.println("Book "+(i+1)+"\n");
System.out.print("Enter name of book: ");
name = Scanner.next();
System.out.print("Enter author: ");
author = Scanner.next();
price = Scanner.nextDouble();
System.out.print("Enter the number of pages: ");
num_page = Scanner.nextInt();
Book b[] = new Book(name, author, price, num_page);
for(int i=0; i<n; i++)
    System.out.print("Book "+(i+1)+":\n"+b[i]);
```

Output

```
Enter number of books: 2
Enter name of Book: Tears in the Dofy Breeze
Enter Author of Book: Divakar Babu
Enter Number of pages in Book: 20
Name: Tears in the Dofy Breeze Author: Divakar Babu
Price: 500 No of pages: 20
Enter name of Books: Journey to Eldorado
Enter Author of Book: Divakar Babu
Enter Number of pages in Book: 10
Name: journey to Eldorado Author: Divakar Babu
Price: 500 No of page: 10
```

Q) Java program to create class Student contain USN, Name, marks. It include method to accept details.

import java.util.Scanner;

class Student {

 private String USN;

 private String name;

 private int[] marks;

 public Student(String USN, String name) {

 this.USN = USN;

 this.name = name;

 this.marks = new int[6];

 }

public void acceptDetails() {

Scanner scanner = new Scanner(System.in);

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

this.USN = scanner.nextLine();

for (int i = 0; i < marks.length; i++) {

System.out.print("Enter marks for Subject " +

(i + 1) + ": ");

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

}

}

public double calculatePercentage() {

int totalMarks = 0;

for (int mark : marks) {

totalMarks += mark;

}

return (double) totalMarks / marks.length;

}

public void displayDetails() {

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

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

```
System.out.print("Marks = ");  
for (int i=0; i < marks.length; i++) {  
    System.out.print("Subject" + (i+1) + ":" + marks[i] + " ");
```

8
System.out.println();

```
System.out.println("percentage = " + calculatePercentage());  
System.out.println("Total = " + "100");
```

8
public class StudentArrayListDemo {

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

```
Scanner scanner = new Scanner(System.in);  
int numStudents = scanner.nextInt();
```

```
Student[] students = new Student[numStudents];  
for (int i=0; i < numStudents; i++) {
```

```
System.out.println("Enter details for  
Student " + (i+1) + "...");
```

```
Student s[] = new Student(0, 0, 0);
```

```
students[i] = acceptDetails(s);
```

8
public void displayDetails() {

```
System.out.println("U&N: " + this.usn);
```

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

```
System.out.println("Details of Students:");  
for (Student student : students) {
```

```
student.displayDetails();
```

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

Java code to calculate roots of quadratic equation
 - Input: coefficients a, b, c
 - Output: roots r1, r2

Quadratic root

(3)

```
import java.util.Scanner;
import java.util.Scanner;
import java.lang.Math;
class Quadratic {
    Scanner S = new Scanner(System.in);
    System.out.print("Enter constant in form
    ax^2 + bx + c");
    this.a = S.nextInt();
    this.b = S.nextInt();
    this.c = S.nextInt();
}
```

3

```
void calculateRoot() {
    int d = b * b - 4 * a * c;
    if (d == 0) {
        double root1 = -b / (2 * a);
        System.out.println("Roots are " + root1 + " And " +
                           root1);
    } else if (d > 0) {
        double root1 = (b + Math.sqrt(d)) / (2 * a);
        double root2 = (-b - Math.sqrt(d)) / (2 * a);
        System.out.println("Roots are " + root1 + " And " +
                           root2);
    }
}
```

3

Else if

```
double r1 = -b / (2 * a);
```

```
double r2 = Math.sqrt(d) / 2 * a;
```

```
System.out.println("Roots are " + r1 + " & " +
                   r2 + " And " + r1 + " - " + r2);
```

3

3

class runs

```
public static void main(String args[])
{
    Quadratic q = new Quadratic();
    q.calculateRoots();
}
```

9

Output

Enter constant in form ax²+bx+c
1 -7 10

Roots are 5.0 and 2.0

Enter constant in form ax²+bx+c
1 2 1

Roots are -1.0 and -1.0

Enter constant in form ax²+bx+c
1 2 3

Roots are -1 + i(1.414) and -1 - i(1.414)

Output for 2nd program (student)

Enter # number of Students 1

Enter name of Student : Gururaj

Enter UIN of Student : IBM2215093

Enter marks of Student out of 50

Enter mark for Subject 1 : 40

Enter mark for Subject 2 : 40

Enter mark for Subject 3 : 50

Total percentage is 86.6 %

1 Develop a

named

method

Rectangle

classes

contain

of the g

abstract

protected

protected

public

this . di

this . d

2 public

of

Class Re

public

super

3 public vo

int o

System

2

3

class Tr

public

- sup

3

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 the each one of the classes extends the class Shape. Each one of the classes contain the method printArea() that prints the area of the given shape.

Abstract class Shape {

protected int dimension1;

protected int dimension2;

public Shape (int dimension1, int dimension2) {

this.dimension1 = dimension1;

this.dimension2 = dimension2;

}

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);

}

(2) Develop
main
method
with
parameters

```
public void printArea() {
    double area = 0.5 * dimensions1 * dimensions2;
    System.out.println("Area of Triangle is " + area);
}

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

```
public void printArea() {
    double area = Math.PI * dimension1 * dimension2;
    System.out.println("Area of Circle is " + area);
}
```

```
public class Area {
}
```

```
public static void main(String[] args) {
    Rectangle rectangle = new Rectangle(4, 5);
}
```

```
rectangle.printArea();
```

```
Triangle triangle = new Triangle(3, 6);
```

```
triangle.printArea();
```

```
Circle circle = new Circle(7);
```

```
circle.printArea();
```

Output

Area of Rectangle is 20

Area of Triangle is 9.0

Area of Circle is 153.93804002584985

- 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. the savings account provides compound interest and withdrawal facilities but no cheque book facility. the current account provides cheque book facility but no interest.

Class Bank {

```

public static void main (String [] args) {
    SavingsAccount savingsAccount = new SavingsAccount
        ("John Doe", "SA1001");
    CurrentAccount currentAccount = new CurrentAccount ("Jane
        Smith", "CA 2002");
    savingsAccount.deposit (5000);
    savingsAccount.computeInterest ();
    savingsAccount.displayBalance ();
    savingsAccount.withdraw (2000);
    savingsAccount.displayBalance ();

    currentAccount.withdraw (5000);
    currentAccount.displayBalance ();
}

```

~~Class Account {~~

```

protected String customerName;
protected String accountNumber;
protected double balance;
public Account (String customerName, String accountNumber)
{
    this.customerName = customerName;
    this.accountNumber = accountNumber;
    this.balance = 0;
}

```

this.customerName = customerName;

this.accountNumber = accountNumber;

this.balance = 0;

```
public void deposit (double amount) {  
    balance += amount;  
    System.out.println ("Deposit of $" + amount + " successful");  
}  
  
public void displayBalance () {  
    System.out.println ("Account Number: " + accountNumber  
        + " Balance: " + balance);  
}
```

class SavingsAccount extends Account {
 public SavingsAccount (String customerName, String
 accountNumber) {
 super (customerName, accountNumber);
 }

```
public void computeInterest () {  
    double interestRate = 0.05;  
    double interest = balance * interestRate;  
    balance += interest;  
    System.out.println ("Interest of $" + interest + "  
        computed and added to the balance.");  
}
```

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

Output
Deposit
Balance
Interest
balance

class CurrentAccount extends Account {
private double minimumBalance = 1000;
public CurrentAccount (String customerName, String
accountNumber) {
super (customerName, accountNumber);

}
public void withdraw (double amount) {
if (balance - amount >= minimumBalance) {
balance -= amount;
System.out.println ("withdrawal of " +
amount + " successful.");

} else {

System.out.println ("withdrawal of " +
amount + " successful.");

"Insufficient funds. Service
charge applied.");

imposeServiceCharge();

}

private void imposeServiceCharge () {
double serviceCharge = 20;

balance -= serviceCharge;

System.out.println ("Service charge of \$" +
serviceCharge + " imposed.");

}

Output

Deposit of \$ 5000.0 successful
Account Number: 8A1001

Balance: 5000.0

Interest of \$ 250.0 computed and added to the
balance.

Account Number : SA1001
Balance : 5250.0
withdrawal of 2000.0 successful
Account Number : SA1001
Balance : 3250.0
Deposit of ₦ 8000.0 successful
Account Number : CA2002
Balance : 8000.0
withdrawal of 5000.0 successful
Account Number : CA2002
Balance : 3000.0

8.213 R

~~to identify bank account &
amount & time
account number
date
(1) 8/1/20
(2) 9/1/20~~

8

~~8.213 R
to identify bank account & amount
(1) 8/1/20
(2) 9/1/20
account number - a similiar
(1) 8/1/20
(2) 9/1/20~~

8

~~8.213 R
to identify bank account & amount
(1) 8/1/20
(2) 9/1/20
account number - a similiar
(1) 8/1/20
(2) 9/1/20~~

8

Write a Java program to implement package by creating two SEE & CIE packages which contains Internal.java, Students.java & External.java.

```

import CIE.*;
import SEE.*;
public class Main{
    public static void main (String[] args){
        int [] internalMarks1 = {80, 75, 90, 85, 88};
        Internal student1 = new Internal ("1ABC123", "John Doe", 3, internalMarks1);

        int [] SeeMarks1 = {70, 85, 78, 92, 88};
        External student2 = new External ("2XYZ456", "Jane Smith", 3, SeeMarks1);

        int [] finalMarks1 = new int [5];
        for (int i=0; i<5; i++){
            finalMarks1[i] = Student1.internalMarks[i] +
                Student2.seeMarks[i];
        }
    }
}

```

```

System.out.println("Final Marks for " + Student1.usn +
                    " (" + "USN: " + student1.usn + ")");
for (int i=0; i<5; i++){
    System.out.println("Course " + (i+1) + ":" +
                    finalMarks1[i]);
}

```

?

?

External / SEE

package SEE;

import CIE.Students;

public class External extends Students {

public int[] SeeMarks;

public External (String USN, String name,
int sem, int[] SeeMarks)

Super(USN, name, Sem);

this.SeeMarks = SeeMarks

CIE

package CIE;

public class Internals extends Students {

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

public Internals (String USN, String name, int sem,
int[] internalMarks) {

Super(USN, name, sem);

this.internalMarks = internalMarks

package CIE;

public class Students {

public String USN;

public String name;

protected int sem;

public Students (String USN, String name, int sem)

This.USN = USN;

this.name = name;

This.sem = sem;

Output

Final

Course

Output

Final Marks for null (USN: SABC123)

Course 1: 150

Course 2: 160

Course 3: 168

Course 4: 177

Course 5: 176

88
23/12.02

Printed with coding

(Experiment) when does stack overflow

(calculated max - stack overflow)

(estimated max - max)

limit

: C:\gpa\stud\method

; (loop limit) max

; (soft pointer) limit

; (memory free, memory?)

Printed with coding

; soft limit

; (soft limit) memory

; max - min size

Memory dump, dumping binary

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 less than father's age.

public class Main{

 public static void main(String[] args){

 Father father = new Father(10);

 Son son = new Son(20);

 try{

 father.checkAge();

 son.checkAge();

 } catch (WrongAge e) {

 System.out.println(e);

}

 }

 public class Father{

 int Age;

 Father(int Age){

 this.Age = Age;

 void checkAge() throws WrongAge{

 if (Age < 0){

 throw new WrongAge("Age can't be less than zero");

}

public class Son Extends Father {
 Son(int Age){
 super(Age);
 }

@Override
void checkAge(){
 if (Age > Super.Age){
 throw new WrongAge("Son's age cannot
be greater than father's age");
 }
}

public class WrongAge Extends RuntimeException{
 WrongAge (String msg){
 super(msg);
 }

Output
Exception:
WrongAge: Age can't be less than zero

② Write a program which creates two threads,
One thread displaying "BMS college of Engineering"
Once every ten seconds and another displaying "CSE"
Once every two seconds

Class Bms implements Runnable {

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

```
for (int i=0; i<100; i++) {
```

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

```
try {
```

```
Thread.sleep(10000);
```

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

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

```
}
```

```
}
```

Class Cse implements Runnable {

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

```
for (int i=0; i<100; i++) {
```

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

```
try {
```

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

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

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

```
}
```

```
}
```

public class Main {

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

```
Runnable bmsce = new Bmsc();
```

```
Runnable cse = new Cse();
```

Thread thread1 = new Thread(bmsce);
Thread thread2 = new Thread(cse);
thread1.start();
thread2.start();

Output

BMSCE

CSE

CSE

CSE

CSE

CSE

BMSCE

CSE

CSE

CSE

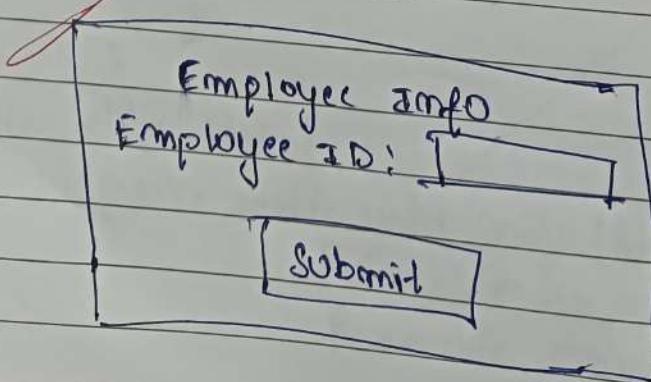
CSE

BMSCE

Don
19/2/24

① creating 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, 20, 80, 30);  
        t.setBounds(20, 100, 80, 30);  
        b.setBounds(100, 100, 80, 30);  
        f.add(l);  
        f.add(t);  
        f.add(b);  
        f.setSize(400, 300);  
    }  
    public void windowClosing(WindowEvent e)  
    {  
        System.exit(0);  
    }  
    public static void main(String args[]){  
        new EventHandling();  
    }  
}
```



② Create a button and add a action Listener for mouse click

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



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



```
from 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();
```

or output



880

2(3/2)