

Pgm.

1. Print "Hello World"

Class Example

```
{  
    public static void main (String a[]){  
        System.out.print("Hello World");  
    }  
}
```

Output:

Hello World.

2. Addition.

Class Example.

{

~~public static void main (String a[]){~~~~{~~

int x=10;

int y=20;

int sum=x+y;

System.out.print(sum);

}

Output:

30

3. Subtraction.

Class Example

```

public static void main(String a[])
{
    int x = 30;
    int y = 20;
    int diff = x - y;
    System.out.print(diff);
}

```

Output: 10.

4. Multiplication.

Class Example

```

public static void main(String a[])
{
    int x = 5;
    int y = 6;
    int prod = x * y;
    System.out.print(prod);
}

```

Output:
30.

5. Division.

class Example

{

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

```
    int x=10, y=5, quotient;
```

```
    quotient = x/y;
```

```
    System.out.print (quotient);
```

{

}

Output:

2

6. check if a number is prime

class Example

{

~~public static void main (String a[])
{~~

```
int n=29, flag=0;
```

```
for (int i=2; i<=n/2; i++)
```

{

```
    if (n%i==0)
```

{

```
        flag=1;
```

```
        break;
```

}

{

```
if (flag==0)
```

{

```
    System.out.println (n + " is a prime");
```

}

```
else
```

```
{
```

```
}
```

System.out.println($n + " is not prime"$)

```
}
```

```
}
```

Output:

29 is prime

7. Fibonacci Series

class Example.

```
{
```

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

```
{
```

int n=10, term1=0, term2=1;

System.out.println("Fibonacci Series till"
+n+"terms");

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

```
{
```

System.out.print(term1+",");

int term3=term1+term2;

term1=term2;

term2=term3;

```
}
```

```
}
```

```
}
```

Output:

Fibonacci Series till 5 terms.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34

2. Method Overloading :

```
class Overload {
    void print(int n) {
        int sum=0;
        for (int i=1; i<n; i++)
        {
            sum = sum + i;
        }
    }
}
```

System.out.print("Sum of "n" natural numbers is "sum");
 }

```
void print(int m, int n) {
```

System.out.println("Prime numbers in the range are ");

```
for (int i=m; i<=n; i++)
{
```

```
    int flag=0;
```

```
    for (int j=2; j<=i; j++)
    {
```

```
        if (i%j == 0)
    {
```

```
            flag = 1;
        break;
    }
```

```
}
```

```
if (flag == 0)
```

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

```
}
```

```
}
```

class OverloadDemo {

public static void main (String [] args)
{

Overload o = new Overload();

o.print(6);

o.print(7, 13);

}

}

Output :

Sum of 6 natural numbers is 2)

Prime numbers in the range are 7, 11, 13.

3. Grocery :

class Grocery {

String cname;

~~String c-ph;~~

~~double total;~~

Grocery (String cname, String c-ph)
{

this.cname = cname;

this.c-ph = c-ph;

}

void calc(double q-dal, double q-pulses,
double q-sugar)

{

total = q-dal * 100 + q-pulses * 80 + q-sugar
* 50; }

void display ()

{

System.out.println("Name" + " " + Phone
number " " + " " + "Total");

System.out.println(c.name + " " + c.ph
+ total);

System.out.println();

}

}

class G Demo {

public static void main (String [] args)

{

Grocery g1 = new Grocery ("Rama", "80905");

Grocery g2 = new Grocery ("Sumith", "73592");

Grocery g3 = new Grocery ("Suhas", "63592");

g1.calc(2, 2, 1);

g1.display();

g2.calc(3, 5, 2);

g2.display();

g3.calc(1, 1, 0.5);

g3.display();

}

}

Output:

Name	Phone number	Total
Rama	80905	410.0

Name	Phone number	Total
Sumith	73592	800.0

Name	Phone number	Total
Suhas	63592	255.0

3. Quadratic Equation.

```

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 imaginary = Math.sqrt(-d) / (2 * a);
```

~~or~~

```
System.out.println("The equation has  
two complex roots: " + real + " + " +  
"i" + " " + imaginary + " ; and " +  
real + " " + "i" + " " +  
imaginary + "i");
```

```
}
```

```
}
```

```
class Main {
```

```
public static void main(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:

415

Enter b: 12

Enter c: 654

No ~~real~~ real solutions. Roots are imaginary.

~~The Equation has two complex roots:~~

~~0.0 + 1.255266i and 0.0 - 1.255266i~~

8/12/24

```
class Run {
    public static void main(String args[])
    {
        Scanner scanner = new Scanner(System.in);
        int n;
        System.out.print("Enter number of books:");
        n = scanner.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 the book:");
            String name = scanner.next();
            System.out.print("Enter author name:");
            String author = scanner.next();
            System.out.print("Enter the price:");
            int price = scanner.nextInt();
            System.out.print("Enter number of pages:");
            int numPages = scanner.nextInt();
            b[i] = new Books(name, author, price, numPages);
        }
        for (int i=0; i<n; i++)
        {
            System.out.println("Book " + (i+1) + ": " + b[i]);
        }
    }
}
```

2. Write a Java program to create a class Student with members USN, name, marks (6 subjects). Include methods to accept student details and marks. Also include a method to calculate the percentage and display appropriate details (Array of student object to be created).

=>

```
import java.util.Scanner;  
class Student {  
    String USN;  
    String name;  
    double [] marks = new double[6];  
  
    void inputDetails(){  
        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 marks for 6  
        subjects:");  
        for (int i=0; i<6; i++) {  
            System.out.print("Subject" + (i+1) +  
                ":");  
            marks[i] = scanner.nextDouble();  
        }  
    }  
}
```

```
ble calculatePercentage()
{
    double totalMarks=0;
    for (double mark:marks)
    {
        totalMarks += mark;
    }
    return (totalMarks / 6);
}

void displayDetails()
{
    System.out.println ("n Student Details:");
    System.out.println ("n USN "+ USN);
    System.out.println ("n Name "+ name);
    System.out.println ("n Percentage "+ calculate
        Percentage () + "%");
}

public class Main {
    public static void main (String [] args)
    {
        Scanner scanner = new Scanner (System.in);
        System.out.print ("Enter no of Students ");
        int number of Students = scanner.nextInt();
        Student [] students = new Student [number of
            Students];
        for (int i=0; i<number of Students; i++)
        {
            S.O.P ("Enter details of Student "+(i+1));
            students [i] = new Student ();
            students [i].inputDetails ();
        }
    }
}
```

```
for (Student student : students)
```

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

```
}  
}
```

Output:

Enter number of Students : 2

Enter details for student 1 :

Enter USN : 308

Enter Name : Tejas

Enter marks for 6 subjects :

Subject 1 : 45

Subject 2 : 56

Subject 3 : 59

Subject 4 : 45

Subject 5 : 12

Subject 6 : 35

Enter details for Student 2 :

Enter USN : 307

Enter Name : Taun

Enter marks for 6 subjects :

Subject 1 : 35

Subject 2 : 32

Subject 3 : 65

Subject 4: 98
Subject 5: 45
Subject 6: 15

Student Details:

USN: 308

Name: Tejas

Percentage: 47.66

Student Details:

USN: 307

Name: Tarun

Percentage: 57.33

Input for Book DataBase

Enter number of books: 2

Book 1:

Enter name of book: Book 22

~~Enter author name: Suhars~~

Enter price of book: 300

Enter number of pages of book: 100

Book 2:

~~Enter name of book: Book 11~~

~~Enter author name: Tarun~~

Enter price of book: 250

Enter number of pages of book: 350.

haby

PAGE NO:
DATE: 19/01/2024

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.

→
~~abstract~~ class Shape {
 int x;
 int y;
 public abstract void printArea(); } } }

class Rectangle extends Shape {
 printArea(int n, int y)
 {
 int Area = x * y;
 S.O.P("Area" + Area);
 } } }

class Triangle extends Shape {
 printArea(int height, int breadth)
 {
 int Area = $\frac{1}{2} \times x \times y$;
 } } }

S.O.P("Area of triangle" + Area);
}

{

abstract class Shape {

int x;

int y;

public abstract void printArea();

}

class Rectangle extends Shape {

public Rectangle(int length, int breadth)
{ super.length, width);

public void printArea()

int area = x * y;
S.O.P("Area of Rectangle" + area);

class Triangle extends Shape {

public Triangle(int height, int base)

{ super.length, height, base);

public

void printArea()

int Area = 0.5 * height * x * y;
S.O.P("Area of Triangle" + Area);

}

class Circle extends Shape {

public Circle(int radius)

```

    {
        super(radius);
    }
    public void printArea()
    {
        int Area = 3.14 * x * x;
        S.O.P("Area of Circle" + Area);
    }
}

```

```

public class Main {
    public static void main(String args[])
    {
        Rectangle rectangle = new Rectangle(5, 8);
        rectangle.printArea();
        Triangle triangle = new Triangle(10, 10);
        triangle.printArea();
        Circle circle = new Circle(10);
        circle.printArea();
    }
}

```

~~Output:~~

Area of Rectangle : 40
 Area of Triangle : 50
 Area of Circle : 314

Q. Develop a Java Program to create a class Bank having 2 kinds of account for customers, Savings Current.
 Create class Account → acc-name accno, acc-type.

From this derive classes Cur-Acc and Sav-Acc to make them more specific to their requirement.

→ Import java.util.Scanner;

class Account {

}

String customerName;

long accountNumber;

String accountType;

double balance;

public Account(String customerName,

long accountNo, String accountType,

double balance)

{

this.customerName = customerName;

this.accountNumber = accountNumber;

this.accountType = accountType;

this.balance = balance;

}

public void displayBalance()

{

S.O.P("Account Number" + accountNumber);

S.O.P("Customer Name" + customerName);

S.O.P("Account type" + accountType);

S.O.P("Balance" + balance);

}

public void deposit (double amount)

{
balance += amount;

S.O.P ("Deposit of \$" + amount + "Successful");
display Balance();

?
}

public void withdraw (double amount)

{
if (amount <= balance)

{
balance -= amount;

S.O.P ("Withdrawal of \$" + amount + "
Successful");

if else {
?
}

S.O.P ("Insufficient balance");

display Balance();

?
}

class Current extends Account

{
private double minBalance = 1000;

private double servicecharge = 50;

public Current (String customerName,
long accountNo., double balance);
{
super(customerName, accountNo., balance);
}

super(customerName, accountNo., "Current",
balance); }
}

public void withdraw (double amount)

{
if (amount <= balance - minBalance)

{
balance -= amount;

S.O.P ("Withdrawal of \$" + amount + "Successful");
}

if else {

S.O.P("Insufficient fund")
"Service charge" + Service charge +
"imposed");
balance -= Service charge; }
display Balance();
}

class SavAct extends Account

{

private double interestrate = 0.05;
public SavAct(String CustomerName,
long accno, double balance);
{

Super(CustomerName, accno, "Savings",
balance);

public void computeInterest()

{

double interest = balance * interestrate;
balance += interest;

S.O.P("Interest of \$" + interest + "
credited"));

display Balance();

}

class prog 2

{

psvm(String args[])

{

Scanner S = new Scanner(System.in);

SavAct SA = new SavAct("Albert", 1234, 5000);

SA.displayBalance();

SA.deposit(1000);

SA.computeInterest();

SA.withdraw(2000);

CurrAct CA = new CurrAct("Alfred", 98321, 1500);

CA.displayBalance();

CA.deposit(500);

CA.withdraw(1000);

CA.close();

}

}

1. Create a package CIE which has two classes - Student and Internals. The class Student has members like usn, name, sem. The class Internals derived from student has an array that stores the internal marks scored in five courses of current semester of student. Create another package SEE which has 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 sem. Import 2 packages in a file that declares the final marks of student in all five courses.

⇒) package CIE;

```
public class Student {
    protected String usn;
    protected String name;
    protected int sem;
    public Student(String usn, String name,
                  int sem)
}
```

this.usn = usn;

this.name = name;

this.sem = sem;

}

}

public class Internals extends Student

protected int[] internalmarks;

public internalmarks(String USN, String name,
int sem, int[] internal
marks)

{

super(USN, name, sem);

this.internalmarks = internalmarks;

}

package SEE;

import CIE.Student;

public class External extends Student {
protected int[] seemarks;

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

{

super(USN, name, sem);

this.seemarks = seemarks;

}

}

import SEE.External;

import CIE.External;

public class Main {

psvm(String[] args)

{

int n = 10;

~~int[] Internalmarks =~~

Internalmarks[] InternalArray = new Internalmarks[n];

External[] ExternalArray = new External[n];

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

```
internalArray[i] = new Internal();
internalArray[i].usn = "IMS17CS001";
internalArray[i].name = "John Doe";
internalArray[i].sem = 5;
```

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

```
    internalArray[i].internalMarks[j] = 80+j;
}
```

```
externalArray[i] = new External();
externalArray[i].usn = "IMS17CS001";
externalArray[i].name = "John Doe";
externalArray[i].sem = 5;
```

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

{

```
    externalArray[i].externalMarks[j] = 70+j;
}
```

{

}

2. 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.

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

```
class WrongAge extends Exception {
    wrongAge(String message) {
        super(message);
    }
}
```

```
class Father {
```

```
    int age;
    Father(int age) throws WrongAge {
```

```
        if (age < 0) {
```

```
            throw new WrongAge("Invalid age:  
Age cannot be negative!");
```

```
        } this.age = age;
    }
```

```
}
```

```
class Son extends Father {
```

```
    int sonAge;
```

```
Son(int fatherAge, int sonAge) throws WrongAge {
```

```
    super(fatherAge);
```

```
    if (sonAge  $\geq$  fatherAge) {
```

```
        throw new WrongAge("Invalid age");
```

```
}
```

this. SonAge = SonAge ;
}

3

public class ExceptionInheritanceDemo {

psvm(string args[])

Scanner scanner = new Scanner(System.in);

try

S.o.p ("Enter father's age");

```
int fatherAge = scanner.nextInt();
```

S.O.P ("Enter Sons age");

```
int SonsAge = scanner.nextInt();
```

Son son = new Son(fatherAge, sonAge);

S.O.P ("Father's age:" + son's age);

$$S.D.P\left(\text{"Son's age": } + (\text{son's age}) \right)$$

catch (Wrong e) {

S.O.P ("Exceptions caught" + e.getMessage());

finally {

Scanner closer;

3

3

Q/P : Enter Father's age : 45
Enter Son's age : 20

Enter Son's age : 20

Father's age = 45

Son's age = 20.

3.

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 MessageThread extends Thread {
```

```
    private String message;
    private int interval;
```

```
    public MessageThread (String message, int
                           interval) {
```

```
        this.message = message;
```

```
        this.interval = interval;
```

```
}
```

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

```
        try {
```

```
            while(true) {
```

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

```
                Thread.sleep(interval * 1000);
```

```
}
```

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

```
    e.printStackTrace();
```

```
}
```

```
}
```

```
public class DisplayMessages {
```

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

```
        MessageThread thread1 = new MessageThread
```

```
            ("BMS College of Engg", 10);
```

```
        MessageThread thread2 = new MessageThread
```

```
            ("CSE", 2);
```

```
        thread1.start();
```

```
        thread2.start();
```

```
}
```

```
}
```

D/p : CSE

BMS college of Engg

CSG

CSE

CSE

BMS college of Engg

CSE

CSE

CSE

CSE

BMS college of Engg

CSE

CSE

CSG

BMS college of Engg

CSE

:

:

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 Text Field();

l.setBounds(20, 50, 80, 30);

t.setBounds(20, 100, 80, 30);

b.setBounds(100, 100, 80, 30);

f.add(b);

f.add(d);

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:

AWT Example

- X

Employee Id :

Submit

2. Examples:

```
import java.io.*;  
public class ByteArray_ex {  
    public static void main(String args[]) throws Exception {
```

```
        FileOutputStream font1 = new FileOutputStream  
            ("Example.txt");
```

```
        FileOutputStream font2 = new FileOutputStream  
            ("Example2.txt");
```

```
        ByteArrayOutputStream bout = new ByteArrayOutputStream();
```

```
        bout.write(65);
```

```
        bout.writeTo(font1);
```

```
        bout.writeTo(font2);
```

```
        bout.flush();
```

```
        bout.close();
```

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

```
} }
```

O/p :

Success

3. Example 2:

```

import java.io.*;
public class ByteArrayInput {
    public static void main(String[] args)
        throws IOException {
        byte[] buf = {35, 36, 37, 38};
        ByteArrayInputStream byt = new ByteArrayInputStream(
            buf);
        int k = 0;
        while ((k = byt.read()) != -1) {
            char ch = (char) k;
            System.out.println("ASCII value of character is: " + k);
            System.out.println("Special character is: " + ch);
        }
    }
}
  
```

O/p :

ASCII value of character is: 35 Special character is #

ASCII value of character is: 36 Special character is \$

ASCII value of character is: 37 Special character is %

ASCII value of character is: 38 Special character is &

4. 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 {
    Frame f;
    TextField tf;
    EventHandling() {
        f = new Frame();
        f.addWindowListener(this);
        tf = new TextField();
        tf.setBounds(60, 50, 120, 20);
        Button b = new Button("Click me");
        b.setBounds(100, 120, 80, 30);
        b.addActionListener(this);
    }
}
```

```
t = new Frame();
f.addWindowListener(this);
tf = new TextField();
tf.setBounds(60, 50, 120, 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 String args[] {
    new EventHandling();
}}
```

O/p :

Welcomechat me

Q. Example 3:

```

public class FileEx {
    public static void main(String a[])
        throws IOException {
        FileInputStream fin = new FileInputStream
            ("Example.txt");
        int content;
        S.O.P ("Remaining bytes that can be
            read : " + fin.available ());
        content = fin.read ();
        S.O.P (char) content + " ");
        S.O.P (content + " ");
        S.O.P ("Remaining bytes that can be
            read : " + fin.available ());
        S.O.P ("Remaining bytes that can be read : "
            + fin.available ());
    }
}

```

O/p:

6. Example 4

```
import java.io.FileInputStream;
import java.io.IOException;

public class FileEx2 {
    public void (String a[]) throws IOException {
        FileInputStream fin = new FileInputStream
            ("Example.txt");
        byte[] bytes = new byte[20];
        int i;
        char c;
        i = fin.read(bytes);
        S.O.P("Number of bytes read: " + i);
        S.O.P("Bytes read: ");
        for (byte b : bytes) {
            c = (char) b;
            S.O.P(c);
        }
    }
}
```