

⇒ Hello World

class Hello  
{

    public static void main (String args[])  
    {

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

}

}

O/p:

Hello World

⇒ Calculator

import java.util.\*;  
class Calc

{

    public static void main (String args[])  
    {

        Scanner sc = new Scanner (System.in);

        System.out.println ("Enter first no.");

        int a = sc.nextInt();

        System.out.println ("Enter second no.");

        int b = sc.nextInt();

```
System.out.println ("Sum = " + (a+b) +
    "\nDifference = " + (a-b) + "\nProduct
    = " + (a*b) + "\nQuotient =
    + (a/b));
```

{}

{}

O/p:

Enter first no

5

Enter second no

67

Sum = 72

Difference = -62

Product = 335

Quotient = 0

⇒ Palindrome or not

```
import java.util.*;
class Palindrome {
    }
```

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

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

```
System.out.println ("Enter a number");  
int n = sc.nextInt();  
int temp = n;  
int rev = 0;  
while (n > 0)  
{
```

```
    int r = n % 10;  
    rev = rev * 10 + r;  
    n = n / 10;
```

{

```
if (rev == temp)
```

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

```
else
```

```
    System.out.println ("Not palindrome");
```

{

{

O/p :-

Enter a number

676

Palindrome

⇒ Fibonacci series upto n

```
import java.util.*;  
class Fibonacci  
{
```

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

⇒

```
Scanner sc = new Scanner (System.in);  
System.out.println ("Enter a number");  
int n = sc.nextInt();  
int a = 0;  
int b = 1;  
System.out.print (a + " " + b + " ");  
for (int i = 1; i <= n; i++)  
{
```

```
    int c = a + b;
```

```
    System.out.print (c + " ");
```

```
    a = b;
```

```
    b = c;
```

```
}
```

~~/~~

```
}
```

O/p:  
Enter a number

6  
0 1 1 2 3 5 8 13

⇒ Prime number or not

import java.util.\*;  
class Prime

{  
public static void main (String args[])

Scanner sc = new Scanner (System.in);  
System.out.println ("Enter a number");  
int n = sc.nextInt();  
int flag = 0;  
for (int i = 2; i <= n/2; i++)

{  
if (n % 2 == 0)

{  
flag = 1; break;

{  
if (flag == 0)

System.out.println ("Prime");

else

System.out.println("Not ~~prime~~ prime");

3

3

O/p:

Enter a number

7

Prime

8

8/12/23

⇒ Overload the method print that gives sum of n natural numbers when one variable is passed, and gives prime numbers in a range when two parameters are passed.

class Overload

{

void print (int n)

{

    int sum=0;  
    for (int i=1; i<=n; i++)

{

        sum+=i;

}

    System.out.println ("Sum = " + sum);

}

void print (int a, int b)

{

    System.out.println ("Prime numbers are -");

    for (int i=a; i<=b; i++)

{

        int flag=0;

P.T.O →

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

{ if (i % j == 0)

flag = 1;  
break;

{

{

if (flag == 0)

System.out.println(i);

{

{

import java.util.\*;  
class Run

{

public static void main (String args[])

{

Scanner sc = new Scanner(System.in);

Overload obj=new Overload();

System.out.println ("Enter n");

int N = sc.nextInt();

System.out.println ("Enter range");

int A = sc.nextInt();

int B = sc.nextInt();

```
obj.print(N);  
obj.print(A, B);
```

{

3

O/p:

Enter n

5

Enter range

7

13

Sum = 15

Prime numbers are -

7

11

13

⇒ Write a program to create a class `Grocery` that has the variables `c-name` and `c-phone`. Create a method to accept 3 parameters to specify quantity of dal, quantity of pulses, and quantity of sugar. Return total price. Display details of 3 customers.

```
class Grocery
```

```
{
```

```
String c-name, c-ph;
```

```
double total;
```

```
Grocery(String c-name, String c-ph)
```

```
{
```

```
this.c-name = c-name;
```

```
this.c-name
```

```
this.c-ph = c-ph;
```

```
}
```

```
void calc(double q-dal, double q-pulses,
```

```
double q-sugar)
```

```
{
```

$$\text{total} = \text{q-dal} * 100 + \text{q-pulses} * 80 \\ + \text{q-sugar} * 50;$$

```
}
```

```
void display()
```

```
System.out.println("Name\tPhone  
number\tTotal");
```

```
System.out.println(c-name + "\t" +  
c-ph + "\t" + total + "\t");
```

{

{

```
class Run
```

{

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

{

```
Grocery g1 = new Grocery("Rama,"  
"123456789");
```

```
Grocery g2 = new Grocery("Shama,"  
"123456789");
```

```
Grocery g3 = new Grocery("Bhama,"  
"123456789");
```

```
g1.calc(2, 2, 1);
```

```
g1.display();
```

```
g2.calc(3, 5, 2);
```

```
g2.display();
```

```
g3.calc(4, 1, 0.5);
```

```
g3.display();
```

{

{

O/p:

Name	Phone number	Total
Rama	123456789	410.0
Name	Phone number	Total
Shama	123456789	800.0
Name	Phone number	Total
Bhama	123456789	205.6

⇒ Calculate roots of a quadratic equation.  
Use appropriate methods to  
take input & calculate roots.

```
import java.util.*;  
class Quad  
{
```

```
    int a, b, c;  
    double root1, root2, d;  
    Scanner s = new Scanner(System.in);  
    void input()  
    {
```

```
        System.out.println("Enter a, b, c");
```

```
        a = s.nextInt();
```

```
        b = s.nextInt();
```

```
        c = s.nextInt();
```

```
        if (a == 0)
```

```
            System.out.println("Invalid  
Input");
```

```
}
```

```
void discriminant()  
{
```

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

```
}
```

void calculateRoots()

{

if ( $d > 0$ )

{

System.out.println ("Real & Unequal");

root1 =  $(-b + \sqrt{d}) / (2*a)$ ;

root2 =  $(-b - \sqrt{d}) / (2*a)$ ;

System.out.println ("root1 = " + root1 + " and root2 = " + root2);

}

else if ( $d == 0$ )

{

System.out.println ("Real & Equal");

root1 =  $(-b + \sqrt{d}) / (2*a)$ ;

System.out.println (root1);

}

else

{

System.out.println ("No real solutions.");

Roots are imaginary);

double real =  $-b / (2*a)$ ;

double imaginary =  $\sqrt{-d} / (2*a)$ ;

System.out.println ("The equation has

two complex roots : " + real +

" + " + imaginary + " ; and " +

real + " - " + imaginary + "; );

{

{

{

class Main {

{

public static void main (String args [])

Quad q = new Quad();

q . input();

q . discriminant();

q . calculateRoots ();

{

{

6/p :-

Enter a, b, c

5

4

2

No real solutions. Roots are  
imaginary.

the equation has two complex roots:  
0.0 + 0.4898979i ... and 0.0 - 0.4898979i;

12/1/24

Lab 3

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

⇒ Write a 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 and display appropriate details

```
import java.util.*;
class Student
{
    private String usn;
    private String name;
    int marks[] = new int[6];
    Scanner sc = new Scanner(System.in);
    void accept()
```

System.out.println("Enter the USN  
of student:");

usn = sc.next();

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

name = sc.next();

System.out.println("Enter the marks  
of student in all 6 subjects");

```
for (int i=0; i<6; i++)
{
    marks[i] = sc.nextInt();
}
```

```
void calc()
{
```

```
    double perc = 0.0;
    for (int i=0; i<6; i++)
    {
        marks[i] = sc.
        perc += marks[i];
    }
}
```

```
perc /= 6;
```

```
System.out.println("Name: " + name +
    "\n USN: " + usn + "\n"
    "Percentage: " + perc + "%");
```

```
}
```

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

```
    Student obj[] = new Student[10];
    obj.accept();
    for (int i=0; i<10; i++)
        obj[i].calc();
```

```
    }
```

```
    obj[i].accept();
    obj[i].calc();
```

```
}
```

```
}
```

```
}
```

O/p :-

Enter the USN of student

IBM22CS287

Enter the name of student

Sparsha Kadaba

Enter the marks of student in all 6 subjects

78

56

90

99

79

89

Name : Sparsha Kadaba

USN : IBM22CS287

Percentage : 81.833333333333%

// Repeat 9 more times.

⇒ Create a class Book that contains four members: name, author, price and numPages. 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. Develop a java program to create n book objects

```
import java.util.*;  
class Books  
{  
    String name;  
    String author;  
    int price;  
    int numPages;  
    Books(String n, String a, int p, int n)  
    {  
        name = n;  
        author = a;  
        price = p;  
        numPages = n;  
    }  
}
```

String toString()  
{  
 name  
}

String toString()  
{

String name, author, price, numPages;  
name = "Book name:" + this.name + "\n",  
author = "Author:" + this.author + "\n",  
price = "Price: Rs." + this.price + "\n",  
numPages = "No of pages:" + "this.numPages  
+ "\n";

return name + author + price +  
numPages;

{

class Main

{

public static void main(String args[])

Scanner sc = new Scanner(System.in);

```
int n;
```

```
String name;
```

```
String author;
```

```
int price;
```

```
int numPages;
```

```
System.out.println("Enter number of  
books");
```

```
n = sc.nextInt();
```

```
Books b[] = new Books[n];
```

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

~~```
System.out.println("Name :");
```~~~~```
name = sc.next();
```~~~~```
System.out.println("Author :");
```~~~~```
author = sc.next();
```~~~~```
System.out.println(" Price :");
```~~~~```
price = sc.nextInt();
```~~~~```
System.out.println(" No of pages :");
```~~~~```
numPages = sc.nextInt();
```~~

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

3

3

O/p:-

Enter number of books

1

Name:

XYZ

Author:

ABC

Price:

100

No of pages:

90

Book name: XYZ

Author: ABC

Price: Rs. 100

No. of pages: 90

~~Rs. 21~~  
~~Rs. 21~~

⇒ 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 given shape

abstract class Shape  
{

```
    int a, b;
    void printArea();
}
```

class Rectangle extends Shape  
{

```
    void printArea()
    {
```

```
        System.out.println("Area of rectangle = "
            + (a * b));
    }
```

class Triangle extends Shape

```
{  
    void printArea()  
{
```

```
        System.out.println("Area of  
        triangle = " + (0.5 * a * b));
```

{

}

class Circle extends Shape

```
{  
    void printArea()  
{
```

```
        System.out.println("Area of  
        circle = " + (3.14 * a * a));
```

}

{

class Run

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

```
        Rectangle obj1 = new Rectangle();  
        obj1.a = 3; obj1.b = 4;
```

obj1.printArea();

Triangle obj2 = new Triangle();

obj2.a = 5; obj2.b = 6;

obj2.printArea();

Circle obj3 = new Circle();

obj3.a = 9;

obj3.printArea();

3

O/p:

Area of rectangle = 12

Area of triangle = 15.0

Area of circle = 254.34

8/1/24

⇒ Bank program

```
import java.util.*;
class Account
{
```

```
    String customerName;
```

```
    int accNo;
```

```
    String accType;
```

```
    double balance;
```

```
    Account(String name, int no, String
            type, double bal)
```

```
{
```

```
        customerName = name;
```

```
        accNo = no;
```

```
        accType = type;
```

```
        balance = bal;
```

```
}
```

```
    void deposit(double amount)
```

```
{
```

```
        balance += amount;
```

```
        System.out.println("New Balance: " + balance);
```

```
}
```

```
    void display()
```

```
{
```

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

```
}
```

class CurAcc extends Account

{ double minBalance = 1000;

double serviceCharge = 50;

CurAcc (String name, int no, String type, double bal)

{

super (name, no, type, bal);

}

@ Override

void deposit (double amount)

}

super . deposit (amount);

@ Override

void display ()

{

super . display ();

void checkMinBalance()

{ if (balance < minBalance)

balance -= serviceCharge;

System.out.println ("Service charge imposed"); }

}

class SavAcc extends Account  
{

    double interestRate = 0.05;

    SavAcc( String name, int no,

                String type, double bal )

    {  
        super( name, no, type, bal );

    @ Override

    void deposit( double amount )  
{

        super.deposit( amount );  
    }

    @ Override

    void display()  
{

        super.display();  
    }

    void compute()  
{

        double interest = balance + interestRate;

        balance += interest

        System.out.println("New balance :  
                              " + balance);

}

void  
  {

withdraw (double amount)

    if (amount <= balance)  
      {

balance -= amount;

        System.out.println ("New  
                          balance:" + balance);

}

    else  
      {      System.out.println ("Insufficient  
                          balance");

}

}

}

public class Bank

{ public static void main (String [] args)

Scanner sc = new Scanner (System.in);

    SavAcc savings = new SavAcc ("John Doe",  
                          1001, "Savings", 5000);    CurAcc current = new CurAcc ("Jane Doe",  
                          2001, "Current", 3000);

```
System.out.println("Savings Account-");
savings.display();
savings.deposit(2000);
savings.compute();
savings.display();
saving.withdraw(3000);
System.out.println("Current Account-");
current.display();
current.deposit(1000);
current.checkMinBalance();
current.withdraw(4000);
```

{

{

O/p:

Savings Account

Balance : 5000.0

New balance : 7000.0

New balance : 7350.0

Balance : 7350.0

New balance : 4350.0

Current Account-

Balance : 3000.0

New balance : 4000.0

Service charge imposed

New balance : 3950.0

Balance : 3950.0

New balance : -50.0

~~SB~~

Lab 5  
16/2/24

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

⇒ Son and Father exception handling program

class WrongAge extends Exception  
{

    public WrongAge (String msg)  
    {

        super (msg);

    }

}

class Father

{

    int age;

    Father (int age) throws WrongAge

{

    if (age < 0)

    {

        throw new WrongAge ("Age cannot be negative");

    }

    this.age = age;

    }

}

class Son extends Father

{

int sonAge;

Son (int fatherAge, int sonAge) throws  
WrongAge

{

super (fatherAge);

if (sonAge >= fatherAge)

{

throw new WrongAge ("Son's  
age cannot be greater  
than father's age");

{

this.sonAge = son Age;

{

{

class Main

{

public static void main (String [] args)

{ try

Father f = new Father (-50);

Son s = new Son (50, 55);

{

catch (WrongAge e)

{

{

System.out.println (e);

{

{

O/p:-

~~WrongAge: Age cannot be negative.~~

8✓

→ WAP which creates two threads, one thread displaying 'BMS College of Engineering' once every ten seconds and another displaying 'CSE' once every two seconds

public class Main

{ public static void main(String[] args)

    Thread bms = new Thread(new bmsRunnable());  
    Thread cse = new Thread(new cseRunnable());  
    bms.start();  
    cse.start();

}

}

class bmsRunnable implements Runnable

{ @Override

    public void run()

    { while(true)

        {

            System.out.println("BMS");

```
try
{
    Thread.sleep(10000);
}
catch (InterruptedException e)
{
}
}

// end while
}

// end run
}

// end class
```

```
class weRunnable implements Runnable
{
```

@Override

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

```
    while(true)
{
```

```
        System.out.println("(SE)");
    }
}
```

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

```

}
catch (InterruptedException e)
{
```

3//end catch

3//end while

3//end run

3//end class

O/p:-

BMS

CSE

CSE

CSE

CSE

CSE

BMS

CSE

CSE

CSE

CSE

CSE

BMS

## ⇒ Package Program

```
package CIE;
import java.util.Scanner;
```

```
public class Student
```

```
    protected String usn = new String();
    protected String name = new String();
    protected int sem;
    public void input()
```

~~Scanner sc = new Scanner (System.in);  
System.out.println ("Enter name");  
name = sc.nextLine();~~

~~System.out.println ("Enter usn");  
usn = sc.nextLine();~~

~~System.out.println ("Enter sem");  
sem = sc.nextInt();~~

}

```
public void display()
```

~~System.out.println ("Name: " + name +
 "\nUSN: " + usn + "Sem: " + sem);~~

3

3

```
package CIE;
import java.util.Scanner;
public class Internals extends Student
```

```
{ protected int marks[] = new int[5];
  public void input(CIEmarks())
{ }
```

```
Scanner sc = new Scanner(System.in);
System.out.println("Enter marks:");
for (int i=0; i < marks.length; i++)
{ }
```

```
marks[i] = sc.nextInt()
```

{

{

{

```
package SEE;
import CIE.Internals;
import java.util.Scanner;
public class External extends Internals
```

```
{ protected int marks[];
```

```
protected int finalMarks[];
```

```
public External()  
{
```

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

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

{

```
public void inputSEE marks()  
{
```

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

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

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

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

{}

```
public void calcFinal Marks()  
{
```

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

```
    finalMarks[i] = marks[i]/2 + super_marks[i];
```

{}

```
public void displayFinalMarks()  
{
```

```
    display Student Details();
```

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

```
        System.out.println ("Subject" +
```

(i + 1) + ":" + finalMarks[i]);

{

{

import SEE.External;  
class MainMarks

{

public static void main(String args[])

{ int n = 2;

External finalMarks[] = new External[n];

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

finalMarks[i] = new External();

finalMarks[i].inputStudentDetails();

System.out.println("Enter IIE marks");

finalMarks[i].inputIIEmarks();

System.out.println("Enter SEE marks");

finalMarks[i].inputSEEmarks();

{}

System.out.println("Displaying data:\n");

for (for i

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

3

finalMarks[i]. calculateFinalMarks();  
finalMarks[i]. displayFinalMarks();

3

3

~~Q~~

⇒ 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);
```~~

3

```
public void windowClosing(WindowEvent)
```

```
{  
    System.exit(0);  
}
```

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

```
{  
    AWTExample awtobj = new AWTExample();  
}
```

O/p:

~~Employee ID~~

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

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

```
} AWT Example awt obj = new AWTExample();
```

O/p:

~~Employee ID~~

Submit

⇒ Create a 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();  
{
```

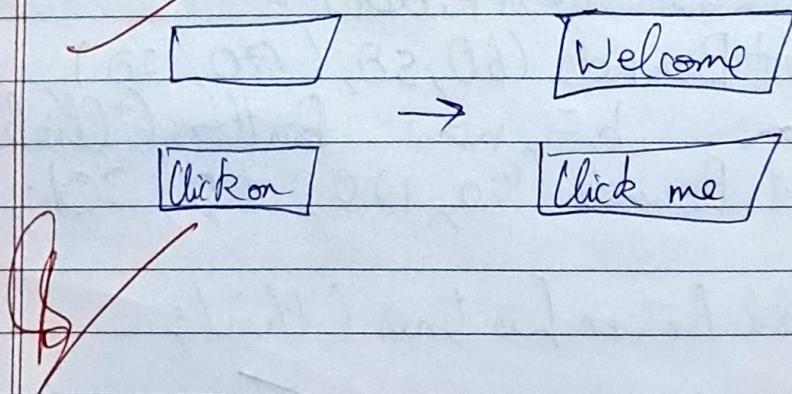
```
f = new Frame();  
f.addWindowListener(this);  
tf = new TextField();  
tf.setBounds(60, 50, 70, 20);  
Button b = new Button("Click me");  
b.setBounds(100, 120, 80, 30);
```

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

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

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

O/p:



## ⇒ Programs on I/O

```

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 (k + " → " + ch);
    }
}

```

O/P.

35 → #

36 → \$

37 → %

38 → 8

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

FileOutputStream fout1 = new

FileOutputStream("Example1.txt");

FileOutputStream fout2 = new

FileOutputStream("Example2.txt");

ByteArrayOutputStream bout

= new ByteArrayOutputStream();

bout.write(65);

bout.writeTo(fout1);

bout.writeTo(fout2);

bout.flush();

bout.close();

System.out.println("Success");

}

O/p:

Success

• public class FileEx

{ public static void main (String a[])  
throws IOException {

FileInputStream fin = new FileInputStream  
("Example.txt");  
byte[] bytes = new byte[20];

int i;

char c;

i = fin.read (bytes);

System.out.println ("No. of bytes read: " + i);

System.out.print ("Bytes read: ");

for (byte b: bytes)

c = (char) b;

System.out.print (c);

}

O/p:

Remaining bytes : 1

Remaining bytes that can be read: 0

```
import java.io.FileInputStream;
import java.io.IOException;
public class FileEx {
    public static void main (String a[])
        throws IOException {
        FileInputStream fin = new FileInputStream("Example.txt");
        byte[] bytes = new byte[20];
        int i;
        char c;
        i = fin.read (bytes);
        System.out.println ("No of bytes read:");
        System.out.print ("Bytes read:");
        for (byte b: bytes) {
            c = (char) b;
            System.out.print (c);
        }
    }
}
```

O/p:

No of bytes read: 20

Byte read: Hello he is good

\$20/20