

1) Print "Hello World"

class Example

{

 public static void main (String args [])

{

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

}

OUTPUT:

Hello World

2) Addition

class Example

{

 public static void main (String args [])

{

 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 = 10, y = 5, diff;
    diff = x - y;
    System.out.print (diff);
}
```

OUTPUT:

5

4) multiplication

class Example

```
{ public static void main (String a[])
{
    int x = 5, y = 6, prod;
    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);
}
```

8/12/23

OUTPUT:

2

6) check if a number is prime

class Example

{

public static void main(String ar[])

{ 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 number");

}

else

{

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

}

}

OUTPUT:

29 is a prime number

7) Fibonacci Series

class Example

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

8) Roots of a quadratic equation

class Example

```
{  
    public static void main (String args[])  
    {  
        double a = 1, b = -7, c = 10;  
        double root1, root2;  
        double determinant = b * b - 4 * a * c;  
        if (determinant > 0)  
        {
```

```

root1 = (-b + Math.sqrt(determinant));
root2 = (-b - Math.sqrt(determinant));
System.out.print("root1 = " + root1 + " and ");
root2 = " + root2);
}

else if (determinant == 0)
{
    root1 = root2 = -b / (2 * a);
    System.out.print("root1 = root2 = " + root1);
}

else
{
    double real = -b / (2 * a);
    double imaginary = Math.sqrt(-determinant) /
        (2 * a);
    System.out.print("root1 = " + real + " + " +
        imaginary + "i");
    System.out.print("root2 = " + real + " - " +
        imaginary + "i");
}
}

```

~~for more on imaginary~~

OUTPUT:

root1 = 2 and root2 = 5

Lab 2

- 1) Write a program to overload the method print that prints the sum of n natural numbers when one variable is passed, and prints the prime numbers in a given range when 2 parameters are passed.

ans) class Overload {

 void print(int n)

{

 int sum = 0;

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

{

 sum = sum + i;

}

 System.out.println("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/2; 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(5);
        o.print(5, 13);
    }
}
```

OUTPUT:

Sum of 5 natural numbers is 15

Prime numbers in the range are:

7

11

13

- 2) Write a Java program to create a class ~~Grocery~~ that has variables `c-name` and `c-phone`. Create a method to accept 3 methods to specify quantity of dal, quantity of pulses and quantity of sugar. The method to return the total price. Display the name, ph-no and total bill of 3 customers.

ans)

```
class grocery
```

```
{ String c-name;
```

```
String c-ph;  
double total;
```

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

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

```
    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);
```

```
}
```

```
}
```

```
class Demo
```

```
{
```

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

```
{
```

~~grocery g1 = new grocery ("Ramu", "1234567890")~~~~grocery g2 = new grocery ("Shweta", "9876543210")~~~~grocery g3 = new grocery ("Bhavna", "246812345")~~~~g1.calc(2, 2, 1);~~~~g1.display();~~~~g2.calc(3, 5, 2);~~

g2. display();

g3. rate(1, 1, 0.5);
 g3. display();

}

OUTPUT:

Name

Rama

Phone number	Total
1234567890	310

Name

Shoma

Phone number	Total
9876543210	640

Name

Brama

Phone number	Total
246812345	185

- 3) Write a Java program to calculate roots of a quadratic equation. Use appropriate methods to take input and calculate the roots.

ans) 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 calculateRoot()
{
    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("root1 is " + root1);
    System.out.println("root2 is " + root2);
}
```

```
else if (d == 0)
{
```

~~```
 System.out.println("Roots are real and equal");
 root1 = -b / (2 * a);
 System.out.println("Root is " + root1);
}
```~~

```
else
{
```

~~```
    System.out.println("Roots are imaginary");
    double real = -b / (2 * a);
    double imaginary = Math.sqrt(d) / (2 * a);
}
```~~

System.out.println("The 2 roots are: " +
 real + "+" + "i" + imaginary + " and " + real + "-" +
 "i" + imaginary);
 }
 }
 }

class main
{

public static void main (String [] args)
{

Quad q = new Quad();

q.input();

q.discriminant();

q.calculateRoots();

}

}

OUTPUT 1:

Quadratic equation is in the form: $ax^2 + bx + c$

Enter a: 1

Enter b: -5

Enter c: -14

Roots are real and unequal

root 1 is -2

root 2 is 7

OUTPUT 2:

Quadratic equation is in the form: $ax^2 + bx + c$

Enter a: 1

Enter b: -4

Enter c: 4

Roots are real and equal

Root is 2

OUTPUT 3 :

Quadratic equation is in the form: $ax^2 + bx + c$

Enter a: 1

Enter b: -4

Enter c: 5

Roots are imaginary

The 2 roots are: $2 + i1$ and $2 - i1$

8/12/24

Lab - 3

- i) 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.)

Ans) import java.util.Scanner;

```
class Student
```

```
{ String usn;
```

```
String name;
```

```
double[] marks = new double[6]
```

```
void inputdetails()
```

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

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

```
usn = s.nextLine();
```

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

```
name = s.nextLine();
```

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

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

```
{
```

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

```
marks[i] = s.nextDouble();
```

```
}
```

```
}
```

```
double calculatepercentage()
```

{

```
    double totalmarks = 0;
```

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

{

```
        totalmarks += marks[i];
```

}

```
    return (totalmarks/6)
```

{

```
void display()
```

{

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

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

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

```
System.out.println("Percentage : " +
```

```
calculatepercentage() + "%");
```

{

```
class main
```

{

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

{

~~Scanner s = new Scanner (System.in);~~

```
System.out.print("Enter no. of students : ");
```

```
int num = s.nextInt();
```

```
Student[] students = new Student [num];
```

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

{

```
System.out.print("Enter the details for student " + (i+1) + "
```

```
students[i] = new Student();
students[i].inputdetails();
}
for (int i=0; i < num; i++)
{
    students[i].display();
}
}
```

OUTPUT:

Enter the number of students : 2

Enter details for student 1 :

Enter USN : 1BM22CS285

Enter name : ABC

Enter marks for 6 subjects :

subject 1 : 78

subject 2 : 87

subject 3 : 90

subject 4 : 99

subject 5 : 79

subject 6 : 80

Enter details for student 2 :

Enter USN : 1BM22CS400

Enter name : XYZ

Enter marks for 6 subjects :

subject 1 : 55

subject 2 : 67

subject 3 : 76

subject 4 : 80

subject 5 : 97

subject 6 : 76

student details :

USN : IBM22CS285

Name : ABC

Percentage : 85.5%.

student details :

USN : IBM22CS400

Name : XYZ

Percentage : 71.1666666666667%.

2) create a class Book that contains four members : name, author, price and num-pages. Include a constructor to set the values for the members. Include methods to set and get the details of the object. Include a toString() method that could display the complete details of the book. Develop a Java program to create n book objects.

ans) import java.util.Scanner;

class Books

{

String name;

String author;

int price;

int num-pages;

Books () { }

Books (String name, String author, int price, int num-pages)

{

this.name = name;

this.author = author;

```

this. price = price;
}
this. num- pages = num- pages;
{
public String toString()
{
String name, author, price, num- pages;
name = "Book name :" + this. name + "\n";
author = "Author name :" + this. author + "\n";
price = "Price :" + this. price + "\n";
num- pages = "No. of pages :" + this. num- pages + "\n";
return (name + author + price + num- pages);
}
}

```

```

class main
{

```

```

public static void main (String args [])
{

```

```

Scanner s = new Scanner (System.in);
int n;
String name;
String author;
int price;
int num- pages;
System.out.println ("Enter no. of books:");
n = s.nextInt();
Books b [];
b = new Books [n];
for (int i = 0; i < n; i++)
{

```

```

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

```

```

System.out.print ("Enter name of book");

```

```

name = s.next();

```

```

System.out.print ("Enter author of the book");

```

```

author = s.next();

```

```

System.out.print("Enter price: ");
price = s.nextInt();
System.out.print("Enter no. of pages: ");
num_pages = s.nextInt();
b[i] = new Book(name, author, price, num_pages);
}
for (int i=0; i<n; i++)
{
    System.out.print("Book " + (i+1) + ": " + b[i]);
}
}

```

OUTPUT:

Enter the no. of books : 2

Book 1 :

Enter the name of the book : Jungle Book

Enter the author of the book : Rudyard Kipling

Enter the price of the book : 1000

Enter the no. of pages : 500

Book 2 :

Enter the name of the book : Harry Potter

Enter the author of the book : J K Rowling

Enter the price of the book : 2000

Enter no. of pages : 1000

Book 1 :

Book name : Jungle Book

Author : Rudyard Kipling

Price : 1000

No. of pages : 500

Book 2 :

Book name : Harry Potter

Author name : J K Rowling

Price : 2000

no. of pages : 1000

8

12/1/29

Lab - 4

- 1) Develop a Java program to create an abstract class named Shape that contains 2 integers and an empty method named printArea(). Provide 3 classes named rectangle, triangle and circle such that 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.

ans) abstract class shape

{

int length;

int width;

shape(int length, int width)

{

this.length = length;

this.width = width; /

}

abstract void printArea();

}

class rectangle extends shape

{

rectangle(int length, int width)

{

super(length, width);

}

void printArea()

{

int area = length * width;

System.out.println("Rectangle area:

+ area);

}

}

```
class Triangle extends Shape
```

```
{
```

```
    Triangle (int length, int width)
```

```
{
```

```
    super (length, width)
```

```
{
```

```
    void printArea()
```

```
        double area = 0.5 * length * width;
```

```
        System.out.println ("Triangle area:" +
```

```
        area);
```

```
}
```

```
}
```

```
class Circle extends Shape
```

```
{
```

```
    Circle (int length)
```

```
{
```

```
    super (length, 0);
```

```
{
```

```
    void printArea()
```

~~double area = Math.PI * length * length;~~~~System.out.println ("circle area:" +~~~~area);~~

```
}
```

```
}
```

```
class Main
```

```
{
```

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

```
{
```

```
    Rectangle r1 = new Rectangle (5, 10);
```

```
    r1.printArea();
```

```
    Triangle t1 = new Triangle (4, 6);
```

```

    t.printArea();
    while (c > newCircle(3));
    c.printArea();
}
}

```

OUTPUT:

Rectangle area : 50
 Triangle area : 12
 Circle area : 28.26

✓ 19/1/24

2) Develop a Java program to create a class Bank that maintains 2 kinds of account for its customers, one called savings account provides compound interest and withdrawal facilities but no que book facility but no interest, current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class account that stores customer name, account no., and type of account. From this derive the class withdraw and save-out to make them more specific to their requirements. Include the necessary methods in order to do the following tasks:-

- (i) display balance
- (ii) compute and deposit interest
- (iii) permit withdrawal and update balance

check for minimum balance, impose
penalty if necessary and update the balance

1) class Account

{

```
String customer-name;
int accno;
```

```
String acctype;
double balance;
```

```
Account (String customer-name, int accno,
```

```
String acctype, double balance)
```

{

```
this.customer-name = customer-name;
```

```
this.accno = accno;
```

```
this.acctype = acctype;
```

```
this.balance = balance;
```

}

void deposit (double amount)

{

```
balance += amount;
```

```
System.out.println ("Deposit successful.
```

```
updated balance = " + balance);
```

}

void display()

{

```
System.out.println ("Account no: " + accno);
```

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

}

}

class current extends Account

{

```
double minbalance;
```

```
double servicecharge;
```

deposit (String customer-name, int accno,
double balance, double minbalance, double
servicecharge)

{
super (customer-name, accno, balance);
this. minbalance = minbalance;
this. servicecharge = servicecharge;

}

void withdraw (double amount)

{
if ((balance - amount) >= minbalance)

{
balance -= amount;
System.out.println ("Withdrawal
successful. Updated balance: " + balance);
}
else

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

void deckminbalance()

{
if (balance < minbalance)
{
balance -= servicecharge;
System.out.println ("Your balance
not maintained. Service charge imposed");
System.out.println ("Updated balance:
balance");
}
}

class Savant Bank Account

{ double interestrate;

Savant ("String customername, double amount,
double balance, double interestrate")

{ super("customername, accountnumber, "Savant",
balance);

 this.interestrate = interestrate;

{ void computeinterest()

 double interest = balance * (interestrate / 100);
 balance += interest;

 System.out.println("Interest computed,"
 "updated balance: " + balance);

}

{ void withdraw(double amount)

 if (balance >= amount)

 balance -= amount;

 System.out.println("withdrawal

 successful. Updated balance: " + balance);

}

 else

{

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

}

}

}

{ class Bank

```
public static void main (String args[])
{
    Scanner s = new Scanner (System.in)
```

System.out.println ("Enter name, account no, balance, minimum balance, service charge")
 name : s.next();

int accno : s.nextInt();

double balance : s.nextDouble();

double minbalance = s.nextDouble();

double servicecharge = s.nextDouble();

Savings savings = new Savings (name,

accno, balance, minbalance, servicecharge);

savings.deposit(1000);

savings.display();

savings.computeinterest();

savings.withdraw(500);

}

OUTPUT:

Enter name, account no., balance, minimum balance,
 service charge.

John

123456784

2000

500

20

Report successful . updated balance : 3000

Account number: 123456784

Balance : 3000

Interest computed

Withdrawal successful . updated balance : 2700

Lab - 5

- 1) Create a package (IE) which has 2 classes - student and intervals. The class student has members like name, marks, sem. The class intervals derived from student has an array that stores the intervals marks scored in 5 courses of the current semester of the student. Create another package SEE which has one class student which is a derived class of student. This class has an array that stores the SEE marks scores in 5 courses of current sem of student. Import 2 packages in a file that declares the final marks of n students in all 5 courses.

n)

package (IE);

public class Person

{

public String name;

public String name;

public int sem;

public Person(String name, String name, int sem)
{}

this.name = name;

this.name = name;

this.sem = sem;

{}

{}

package (IE);

public class intervals

{

public int[] intervalmarks;

public intervals ($\text{int} \subset \text{internal marks}$)

This. interwaters: interwaters;

} package SEE

import (IE. *perdon*')

public class external extends person
{

public int(7) seewards;

public external (String name, String name,
~~int~~ int sum, int [] geometry)

super (ver, vorne, Stern),

This - scenario : scenario;

3

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

```
import java.util.Arrays;
```

```
import {IEInternal};
```

import <RE. personnel

import SSE - external ;

public does more

15

public static void main (String args[])

Scanners > new Scanner (system.in);

System.out.println("Enter no. of students
in n = <next>");

intervals[] intervalsFor : new intervals[n]

External [7] externaldata = new External [n]

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

1

System, ext. printem ("Enter detail for student
(+1)");

```

System.out.println("USN:");
String usn = s.nextLine();
System.out.println("Name:");
String name = s.nextLine();
System.out.println("Semester:");
int sem = s.nextInt();
}

System.out.println("Enter internal marks
for 5 subjects:");
int[] internalmarks = new int[5];
for (int j = 0; j < 5; j++) {
}

System.out.println("Subject" + (j + 1));
internalmarks[j] = s.nextInt();
}

```

```

System.out.println("Enter external marks
for 5 subjects:");
int[] externalmarks = new int[5];
for (int j = 0; j < 5; j++) {
}

System.out.println("Subject" + (j + 1));
externalmarks[j] = s.nextInt();
}

```

```

internaldata[i] = new InternalData(usn, name,
sem, internalmark);
externaldata[i] = new ExternalData(usn, name,
sem, externalmark);
}

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

```

```

System.out.println("Student" + (i + 1));
System.out.println("Internal marks:");
arrayToString(internaldata[i].internalmark);

```

```
System.out.println("External marks : " + calculateExternalMarks(externalData[i].externalMarks));
System.out.println("Total marks : " + calculateTotalMarks());}

private static int calculateTotalMarks() {
    int internalMarks, externalMarks;
    int total = 0;
    for (int i = 0; i < internalMarks.length; i++) {
        total += internalMarks[i] + externalMarks[i];
    }
    return total;
}
```

OUTPUT:

Enter no. number of students: 2

Enter details for student 1

Enter USN: 1BM22CS300

Enter name: Avinash

Enter Semesters: 1

Enter internal marks for 5 subjects

Subject 1: 50

Subject 2: 30

Subject 3: 40

Subject 4: 45

Subject 5: 50

Enter external marks for 5 subjects

Subject 1: 45

Subject 2: 40

Subject 3: 50

Subject 4: 50

Subject 5: 40

Enter details for student 2

Enter USN: IBM22CS301

Enter name: Dinesh

Enter Semester: 1

Enter internal marks for 5 subjects

subject 1: 40

subject 2: 30

subject 3: 45

subject 4: 50

subject 5: 45

Enter external marks for 5 subjects

subject 1: 50

subject 2: 40

subject 3: 30

subject 4: 45

subject 5: 50

Student 1:

Internal marks:

External marks:

Total marks ($I_E + S_E$) =

Student 2:

Internal marks:

External marks:

Total marks ($I_E + S_E$)

- 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 uses both

father and son's age and extend an exception
if son's age \geq father's age.

and) public class Father

{ int age;

 father (int age)

{ if (age > 0)

{ }

 throw new illegalargumentexception ("age

{")

 this. age = age;

{ }

} public class Son extends Father

{

 int sonage;

 public Son (int age, int sonage)

{

 super (age);

 if (sonage \geq age)

{ }

 throw new illegalargumentexception ("age
 is not < or father's age");

{ }

 this. sonage = sonage;

{ }

} import java.util.Scanner;

public class main

{

 public static void main (String args [])

```
Scanner >= new Scanner (System.in);  
try {
```

```
    System.out.print("Enter father's age: ");  
    int age = s.nextInt();
```

```
    System.out.print("Enter son's age: ");  
    int sonage = s.nextInt();
```

```
    Son sm = new Son (age, sonage);
```

```
    System.out.print("Father's age: " + sm.age);
```

```
    System.out.print("Son's age: " + sm.sonage);
```

```
} catch (IllegalArgumentexception e) {
```

```
    System.out.print("Exception: " +  
        e.getMessage());
```

```
} s.close();
```

```
}
```

OUTPUT:

Enter father's age:

30

Enter son's age:

12

Father's age: 30

Son's age: 12

3) Write a program which creates 2 threads
one thread displaying "BMS college of Engineering"
once every 10 seconds and another displaying
"CSE" once every 2 seconds

Ans) public class main

{
public static void main (String args[])

{
Thread thread1 = new Thread () {

while (true) {

System.out.println ("BMS college of Engineering");
try

{
Thread.sleep (10000);

}
catch (InterruptedException e)

e.printStackTrace();

}

} ;

Thread thread2 = new Thread () {

while (true) {

{

System.out.println ("CSE");

try

{
Thread.sleep (2000);

}
catch (InterruptedException e)

e.printStackTrace();

} ;

Thread1.start();
Thread2.start();
}

OUTPUT :

BMS college of Engineering

CSE

CSE

BMS college of Engineering

CSE

CSE

CSE

CSE

BMS college of Engineering

Lab 6

- 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(b);
        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 wntObj = new AWTExample ();
```

{

OUTPUT:

Employee ID

- 2) Create a button and add a action listener for mouseclick

```
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, 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);
```

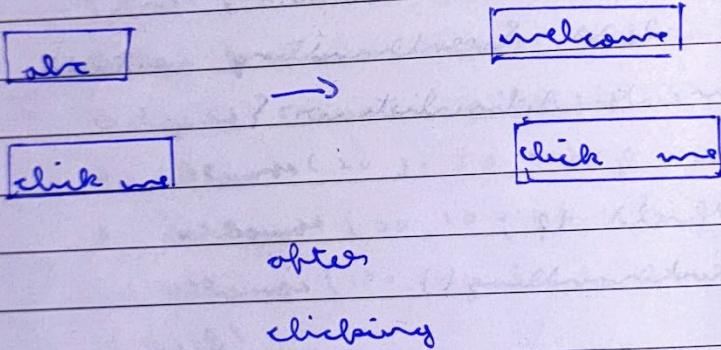
Page

```

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == welcome) {
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
    }
    public static void main(String args[]) {
        new EventHandling();
    }
}

```

OUTPUT



Programs on I/O

```

1) import java.io.*;
public class ByteArrayInput {
    public static void main(String args[]) {
        throws IOException
    }
    byte[] arr = {33, 36, 37, 38};
    ByteArrayInputStream best = new
        ByteArrayInputStream(arr);
}

```

```

int k=0;
while ((a=byte.read()) != -1)
{
    char ch=(char)a;
    System.out.println(" ASCII Value of character
is :" + a + "; special character is ;" + ch);
}
}

```

OUTPUT:

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 :

2) import java.io.*;
 public class Bytearray {
 public static void main (String args[]) throws
 Exception {
 FileOutputStream fout1 = new FileOutputStream
 ("Example.txt");
 FileOutputStream fout2 = new FileOutputStream
 ("Example2.txt");
 Bytearrayoutputstream bout = new Bytearrayoutputstream
 (fout);
 bout.write(65);
 bout.writeTo (fout1);
 bout.writeTo (fout2);
 bout.flush();
 bout.close();
 System.out.println("Success..."); }
 }

INPUT:

Success...

```
3) public class FileEx {
    public static void main (String args[])
        throws IOException {
        FileInputStream fin = new FileInputStream
            ("Example.txt");
        int content;
        System.out.println("Remaining bytes that
            can be read:" + fin.available ());
        content = fin.read ();
        System.out.print((char) content + " ");
        System.out.print("Remaining bytes that
            can be read:" + fin.available ());
        System.out.println("Remaining bytes that can
            be read:" + fin.available ());
    }
}
```

OUTPUT:

Remaining bytes that can be read: 1

A 65 Remaining bytes that can be read: 0

Remaining bytes that can be read: 0

```
4) import java.io.FileInputStream;
import java.io.IOException;
public class FileEx2 {
    public static void main (String args[])
        throws IOException {
        FileInputStream fin = new FileInputStream
            ("Example.txt");
    }
}
```

("Example.txt")

```
byte [] bytes = new byte [20];
```

```
int i;
```

```
char c;
```

```
i = fin.read (bytes);
```

```
System.out.println ("Number of bytes read: " + i);
```

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

```
for (byte i : bytes)
```

```
{
```

```
c = (char) i;
```

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

```
}
```

```
}
```

```
}
```

OUTPUT:

number of bytes read:- 1

Bytes read: A