

08/01/24

PROGRAM-1

Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a , b , c and use the quadratic formula. If the discriminant b^2-4ac is negative, display a message stating that there are no real solutions.

Code :-

```
import java.util.Scanner;  
class Quadratic  
{  
    double a,b,c,d,r1,r2;  
    void calculate()  
{  
        d = b*b - 4*a*c;  
        if(d<0)  
        {  
            System.out.println("Roots are real and distinct");  
            r1 = (-b + Math.sqrt(d)) / (2*a);  
            r2 = (-b - Math.sqrt(d)) / (2*a);  
            System.out.println("Root1 = " + r1 + " and Root2 = " + r2);  
        }  
        else if(d==0)  
        {  
            System.out.println("Roots are real and equal");  
            r1 = -b / (2*a);  
            System.out.println("Root1 = Root2 = " + r1);  
        }  
        else  
            System.out.println("Roots are imaginary");  
    }  
}
```

Class MainRun

(2)

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

```
{ Scanner s = new Scanner (System.in);  
Quadratic obj = new Quadratic ();  
System.out.println ("Enter the value of a");  
obj.a = s.nextDouble();  
System.out.println ("Enter the value of b");  
obj.b = s.nextDouble();  
System.out.println ("Enter the value of c");  
obj.c = s.nextDouble();  
obj.calculate();
```

```
}
```

```
y
```

Output :-

Enter the value of a.

Enter the value of a
1

2

Enter the value of b

Enter the value of b
-4

3

Enter the value of c

Enter the value of c
Roots are real and equal

Roots are imaginary.

Root1 = Root2 = 2.

PROGRAM-2

Write a Java program to create a class Student with members USN, name, marks (6 subjects). Include methods to calculate the percentage and display appropriate details. (Array of student object to be created) (3)

Code:-

```
import java.util.Scanner;  
class Student  
{  
    int usn, i;  
    String name = new String();  
    double marks[] = new double[6];  
    double sum = 0, per;  
    void studentDetails()  
    {  
        System.out.println("Enter student details");  
        Scanner ss1 = new Scanner(System.in);  
        System.out.println("Enter student usn");  
        usn = ss1.nextInt();  
        System.out.println("Enter student name");  
        name = ss1.next();  
        System.out.println("Enter student marks");  
        for (i=0; i<6; i++)  
        {  
            marks[i] = ss1.nextInt();  
        }  
        void display()  
        {  
            System.out.println("Student name "+ name);  
            System.out.println("USN "+ usn);  
            System.out.println("Student marks");  
            for (i=0; i<6; i++)  
            {  
                sum = sum + marks[i];  
            }  
            System.out.println(marks[i] + " ");  
        }  
}
```

```

    per = sum / 120 * 100;
    System.out.println("Percentage = " + per); ④
    sum = 0;
}

class Run
{
    public static void main (String args[])
    {
        Scanner ss2 = new Scanner (System.in);
        System.out.print("Enter total number of students");
        int n = ss2.nextInt();
        Student ss1[] = new Student[n];
        for (int i=0; i<n; i++)
        {
            ss1[i] = new Student();
            ss1[i].student_details();
            System.out.println(i+1 + ". Student details:");
            for (int j=0; j<10; j++)
            {
                ss1[i].display();
            }
        }
    }
}

```

Output:-
enter total number of students

3
enter student details

enter student id

10

enter student name

Rohan

enter student marks

19

20

15

20

16

18

19
enter student details
enter student usn
12
enter student name
Rajesh
enter student marks
15
20
16
20
10
19
;
enter student details
enter student usn
31
enter student name
Ravi
enter student marks
20
15
15
16
20
19

Student details:

student name Rohan

USN 10

student marks

19.0 20.0 25.0 20.0 16.0 18.0

Percentage = 90.000000 student name Rajesh

USN 12

student marks

15.0 20.0 16.0 20.0 10.0 19.0

Percentage = 89.333333 student name Ravi

USN 31

student marks

20.0 15.0 15.0 16.0 20.0 19.0

Percentage = 87.500000

PROGRAM-3

(6)

Create a class Book that contain 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 objects .
 Include a toString() method that could display the complete details of the book . Develop a Java program to create n book objects .

Code:

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

```
class Books
```

```
{
```

```
    String name, author ;
```

```
    int price, numpages ;
```

```
Books () { }
```

```
Books (String name, String author, int price, int numpages)
```

```
{
```

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

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

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

```
    this.numnumpages = numpages ;
```

```
}
```

```
public String toString()
```

```
{
```

```
    String name, author, price, numpages ;
```

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

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

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

```
    numpages = "number of pages: " + this.numpages + "\n";
```

```
    return name + author + price + numpages ;
```

```
}
```

```
}
```

```
class Main
```

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

```
{
```

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

```
    int n, price, numpages, i ;
```

```
    String author, name ;
```

```
    System.out.println ("Enter no number of books") ;
```

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

```

n = s.nextInt();
Books b[] = new Books[n];
for (i=0; i<n; i++) {
    System.out.println("Book " + (i+1) + ": ");
    System.out.print("Enter the name of the book: ");
    name = s.nextLine();
    System.out.print("Enter the author of the book: ");
    author = s.nextLine();
    System.out.print("Enter the price of the book: ");
    price = s.nextInt();
    System.out.print("Enter number of pages of the book: ");
    numPages = s.nextInt();
    b[i] = new Books(name, author, price, numPages);
}
for (i=0; i<n; i++) {
    System.out.println("Book " + (i+1) + ": " + b[i]);
}
}

```

Output:

Enter the number of books

2

Book 1:

enter the name of the book: Harry-Potter 1

enter the author of the book: JK-Rawling

enter price of the book: 1000

enter number of pages of the book: 500

Book 2:

enter the name of the book: Harry-Potter 2

enter the author of the book: JK-Rawling

enter price of the book: 1200

enter number of pages of the book: 600

Book 1:

Book name: Jungle Book Harry-Potter 1

Author name: JK-Rawling

Price: 1000

number of pages: 500

Book 2:

Book name: Tales Harry - Potter 2

Author name: JK-Rooney

Price: 1200

number of pages: 600

Dad
9/1/24

Create a package CIE which has two classes - Internal and External.
 Class Student has members like usn, name, sum. The class
 Internal has an array that stores internal marks scored in 5 subjects of current semester of one student.
 Create another package which has class External which is a derived class of Student. This class has an array that
 stores the SEE marks scored in 5 subjects of the current
 semester. Import the two packages in a file that declares
 the final marks of one student.

(LAB-6)

student.java:-

```
package CIE;
public class Student {
    public String usn, name;
    public int sum;
    public Student (String usn, String name, int sum) {
        this.usn = usn;
        this.name = name;
        this.sum = sum;
    }
}
```

internals.java:-

```
package CIE;
public class Internals extends Student {
    public int m[] = new int[5];
    public Internals (String usn, String name, int sum, int[] m) {
        super(usn, name, sum);
        this.m = m;
    }
}
```

ext.java

```
import SEE;
if CIE.Student;
class External extends Student {
    public int sm[] = new int[5];
    public External (String usn, String name, int sum, int[] sm) {
        super(usn, name, sum);
        this.sm = sm;
    }
}
```

mainclass.java

(10)

```

import java.util.Scanner;
import CTG.Student;
import CTG.Internal;
import CTG.External;
public class mainclass {
    public static void main(String arg[]) {
        int fm = 0;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter no of students: ");
        int n = in.nextInt();
        Internal[] in = new Internal[n];
        External[] em = new External[n];
        Student[] stu = new Student[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Enter detail for student");
            System.out.println("Enter name: ");
            in.nextLine();
            String name = in.nextLine();
            System.out.println("Enter USN: ");
            System.out.println("Enter semester: ");
            in.nextLine();
            in[i].internal_marks = new int[5];
            in[i].external_marks = new int[5];
            System.out.println();
            System.out.println("Enter Marks details: ");
            for (int j = 0; j < 5; j++) {
                System.out.println("Enter internal marks for course " + (j + 1) + " : ");
                in[i].internal_marks[j] = in.nextInt();
                System.out.println("Enter external marks for course " + (j + 1) + " : ");
                in[i].external_marks[j] = in.nextInt();
            }
        }
    }
}

```

Y

```

System.out.println();
stu[i] = new Student(un, name, sem);
in[i] = new Internal(un, name, sem, internal_marks);
em[i] = new External(un, name, sem, external_marks);

```

```

    System.out.println("Enter marks details :");
    for (int i=0; i<n; i++) {
        System.out.println("Enter name : " + stu[i].name);
        System.out.println("Enter USN : " + stu[i].usn);
        System.out.println("Enter sem : " + stu[i].sem);
        System.out.println("Enter internal marks for course " + i + 1 + " : ");
        System.out.println("Enter external marks for course " + i + 1 + " : ");
        fm += stu[i].intmarks + stu[i].extmarks;
    }
    System.out.println("Total internal marks = " + fm);
    System.out.println("Total external marks = " + em);
    System.out.println("Total marks = " + fm + em);
}

```

OUTPUT:-

Enter no. of student : 2
 Enter details for student 1
 Enter name : Kshitiz
 Enter USN : 1234
 Enter semester : 2
 Enter marks details :
 Enter marks details for course 1 : 34
 Enter marks details for course 2 :
 Enter internal marks for course 1 : 34
 Enter external marks for course 1 : 98
 Enter internal marks for course 2 : 39
 Enter external marks for course 2 : 97
 Enter internal marks for course 3 : 33
 Enter external marks for course 3 : 56
 Enter internal marks for course 4 : 33
 Enter external marks for course 4 : 64
 Enter internal marks for course 5 : 40
 Enter external marks for course 5 : 90

Enter details for student 2 :

(2)

Enter Name : Madhurika

Enter USN : 1235

Enter semester : 2

Enter marks details :

Enter internal marks for course 1 : 15

Enter external marks for course 1 : 45

Enter internal marks for course 2 : 17

Enter external marks for course 2 : 50

Enter internal marks for course 3 : 39

Enter external marks for course 3 : 60

Enter internal marks for course 4 : 24

Enter external marks for course 4 : 58

Enter internal marks for course 5 : 25

Enter external marks for course 5 : 50

Final marks details :

student 1

Name : Kshitiz

USN : 1234

Semester : 2

Final marks for course 1 : 132

Final marks for course 2 : 136

Final marks for course 3 : 89

Final marks for course 4 : 93

Final marks for course 5 : 130

student 2

Name : Madhurika

USN : 1235

Semester : 2

Final marks for course 1 : 60

Final marks for course 2 : 67

Final marks for course 3 : 99

Final marks for course 4 : 79

Final marks for course 5 : 75

LAB-1
Develop a Java program to develop an abstract class shape (13) that contains two integers and an empty method printArea(). Promote three classes named Rectangle, Triangle and Circle such that each one of the class extends the class shape. Each one of the class contain only one method printArea().

```
import java.util.Scanner;  
abstract class Shape {  
    int x, y;  
    abstract void area();  
    public static void main (String [] args)  
    {  
        Shape obj1 = new Circle ();  
        obj1.area ();  
        Shape obj2 = new Rectangle ();  
        obj2.area ();  
        Shape obj3 = new Triangle ();  
        obj3.area ();  
    }  
}
```

class rectangle extends shape {
 void rectangle () {

```
        Scanner sc = new Scanner (System.in);  
        System.out.println ("Enter length and breadth");  
        x = sc.nextInt ();  
        y = sc.nextInt ();
```

void area () {
 System.out.println ("area of rectangle " + x * y);
 }
}

class triangle extends shape {

void triangle () {

```
        Scanner sc = new Scanner (System.in);  
        System.out.println ("Enter base & altitude");
```

```
        x = sc.nextInt ();  
        y = sc.nextInt ();
```

(9)

```

void area() {
    System.out.println("Area is : " + x*y*0.05);
}

class Circle extends Shape {
    void circle() {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter radius");
        r= sc.nextInt();
        y= r;
    }

    void area() {
        System.out.println("Area is "+ 3.14*x*x);
    }
}

```

Output

Enter length and breadth : 4 8

Area is 24

Enter base and altitude : 2 2

Area is 2

Enter radius : 2

Area is 25.12.

LAB-07

(15)

Q. Write a program that demonstrates handling of exceptions. Create a base class. In Father class, implement a constructor that takes the age and throws the exception WrongAge() where input age < 0 . In Son class, implement a constructor that uses both father's and son's age and throws exception if son's age is $>$ father's age.

import java.util.*;
class WrongAge extends Exception { }

```
WrongAge (String msg) {  
    super (msg);  
}
```

{

class Father {

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

{

this . age = age;

if (age < 0) {

throw new WrongAge ("Age can't be less than 0");

{

else {

System.out.println ("Father's age verified");

{

{

class Son extends Father {

int sonage;

Son (int age, int sonage) throws WrongAge { }

super (age);

this . sonage = sonage;

if (sonage < 0 || sonage \geq age) {

throw new WrongAge ("Son's age is invalid");

{

else {

System.out.println ("Son's age verified");

{

{

(16)

class Age {

 public static void main (String[] args) {

 Scanner s = new Scanner (System.in);

 int age, sonage;

 System.out.println ("Enter Father's age : ");

 age = s.nextInt();

 System.out.print ("Enter Son's Age : ");

 sonage = s.nextInt();

 try {

 Son son = new Son (age, sonage);

 }

 catch (WrongAge e) {

 System.out.println (e);

 }

 catch (Exception e) {

 System.out.println (e);

 }

Output

Enter Father's Age : 41

Enter Son's Age : 25

Father's Age verified.

Son's Age verified.

Enter Father's Age : -12

Enter Son's Age : 15

Age can't be less than 0

Wrong Age Exception

Enter Father's Age : 24

Enter Son's Age : 42

Father's Age verified

Son's Age is invalid

Wrong Age Exception

Q. Sample program in threads LAB-78

① class Thread1

```
{ public static void main (String ss[]) }
```

```
{ Thread t = Thread.currentThread ();
```

```
System.out.println ("ee CT : " + t);
```

```
try
```

```
{ for (int n=5; n>0; n--) }
```

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

```
Thread.sleep (500);
```

```
}
```

```
}
```

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

```
{ System.out.println ("ee the sleeping thread "
```

```
woken up ");
```

```
}
```

```
}
```

Output

CT : Thread [main, 5, main]

CT : Thread [current main Thread, 5, main]

5

4

3

2

1

② class NewThread implements Runnable

```
{ Thread t;
```

```
NewThread ()
```

```
t = new Thread (this, "NewThread");
```

```
System.out.println ("ee CT : " + t);
```

```
t.start();
```

```
}
```

public void run()

{
try

{ for (int n=5; n>0; n--)

{ System.out.println ("child :" + n);

Thread.sleep (500);

}

}

catch (InterruptedException e)

{ System.out.println ("child Thread Interrupted");

}

System.out.println ("child Thread quitting");

}

class Thread 2

{

public static void main (String ss[])

{

new NewThread();

System.out.println ("Back in main");

try

{ for (int n=5; n>0; n--)

{ System.out.println ("Main Thread
Interrupted");

}

System.out.println ("Main Thread quitting");

}

}

Output

CT Thread [NewThread, 5, main]

Back in main

Child : 5

Main Thread : 5

Child : 4

Main Thread : 4

Child : 3

Child : 2

Main Thread : 3

child : 1
child Thread quitting
Main Thread : 2
Main Thread : 1
Main Thread quitting

(19)

{ write a program that creates 2 threads, one thread displaying "BMS college of Engineering" once every 10 seconds and another displaying "CSE" once every one seconds.

class NewThread implements Runnable

{ class NewThread implements Runnable
Thread t;
NewThread ()

{ new Thread (this, "NewThread"),
System.out.println ("CSE"+t),
start (),

{ public void run ()

{ try

{ for (int n=10; n>0; n--)

{ System.out.println ("CSE"+n),
Thread.sleep (2000),

{ catch (InterruptedException e) {

{ System.out.println ("Child Thread interrupted"),

{ System.out.println ("Child Thread quitting"),

{

class Thread

{ public static void main (String s[]) {
new NewThread ();

System.out.println("Back in Main");

try

{
for (int i = 0; i < 10; i++)

{

System.out.println("BoMoSoCE");

Thread.sleep(10000);

}
catch (InterruptedException e){
e.printStackTrace();
}

{
System.out.println("Main Thread Interrupted");

}
System.out.println("main Thread quitting");

}

Output

CT Thread [NTthread=5, main]

Back in main

BoMoSoCE

CSE

CSE

CSE

CSE

CSE

BoMoSoCE

CSE

CSE

CSE

CSE

CSE

Main Thread quitting
child Thread quitting

Don
28/2/24

Develop a Java program to create a class Bank that maintains 2 kinds of accounts for its customers, one class savings account and one 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. 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 number, and type of account. From this, define classes cur-acct and sav-acct to make them more specific to their requirements.

- (a) Accept deposit from customer and update balance.
- (b) Display the balance.
- (c) Compute and deposit interest.
- (d) Permit withdrawal and update balance.

→ import java.util.Scanner;
class Account {

String CustomerName;
int accountNumber;
String accountType;
double balance;

Account (String name, int accNo, String type, double bal)
{
CustomerName = name;
accountNumber = accNo;
accountType = type;
balance = bal;}

void deposit (double amount) {

balance += amount;
System.out.println ("Deposit of Rs. " + amount
+ " is successful");

void displayBalance() {

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

}

void withdraw (double amount) {
 if (balance - amount >= 0) {

 balance = amount;
 System.out.println ("Withdrawal of Rs. " +
 amount + " successful");

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

}
 class Current extends Account {

 double minimumBalance;
 double serviceCharge;
 Current (String name, int accNo, String type,
 double bal, double minBal, double charge);

}

f.

super (name, accNo, type, bal);

minimumBalance = minBal;

serviceCharge = charge;

{

void withdraw (double amount) {

 if (balance - amount >= minimumBalance)

 balance = amount;
 System.out.println ("Withdrawal of Rs. " +
 amount + " successful");

} else {

 System.out.println ("Insufficient balance for
 withdrawal Service charge of Rs. " +
 serviceCharge + " applied");
 balance = serviceCharge;

}

void checkbook() {

 System.out.println ("Checkbook facilities
 are available and will be sent soon");

}

class SavAcc extends Account { (23)

 double interestRate;
 SavAcc (String name, int accNo, String type,
 double bal, double rate)

 super (name, accNo, type, bal),
 interestRate = rate;

}

 void computeInterest () {
 double interest = balance * (interestRate / 100);
 balance += interest;
 System.out.println ("Interest of Rs. " + interest +
 " added to account");

}

 void checkbook ()

 { System.out.println ("Checkbook facilities are not
 available"); }

}

}

public class Bank {
 public static void main (String args []) {
 currAcc CA = new currAcc ("Mansoor", 123456,
 5000, 1000, 50),
 SavAcc SA = new SavAcc ("Kashif", 654321,
 10000, 5));

 System.out.println ("Current Account Details");
 CA.displayBalance();

 CA.deposit(2000);

 CA.displayBalance();

 CA.withdraw(7000);

 CA.displayBalance();

 CA.withdraw(3000);

 CA.displayBalance();

 CA.checkbook();

 System.out.println ("In Savings Account
 Details");

 SA.deposit(5000);

 SA.displayBalance();

SA. computeInterest(),
SA. displayBalance(),
SA. withdraw(15000),
SA. displayBalance,
SA. checkbook();

f.

Final output of the program
Output = Rs 15750.00

Output

Savings Account Details:-

Account Balance = Rs 10000.00

Deposit of Rs 5000.0 successful

Account Balance = Rs 15000.0

Interest rate of Rs 750.0 added to the account.

Account Balance : Rs 15750.00

Withdraw of Rs 15000.0 successful

Account Balance : Rs 750.0

Check book facilities not available.

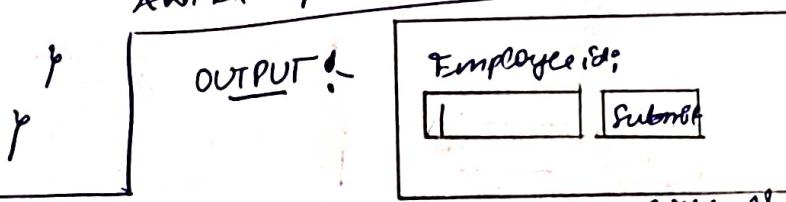
Done
18/11

Q1. Creating label, button and Textfield in a frame using AWT.

```

import java.awt.*;
import java.awt.event.*;
public class AWExample extends WindowAdapter {
    Frame f;
    AWExample() {
        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);
        b.setBounds(20, 100, 80, 30);
        t.setBounds(100, 100, 80, 30);
        f.add(b);
        f.add(l);
        f.add(t);
        f.setSize(400, 300);
        f.setTitle("Employee Info");
        f.setLayout(null);
        f.setVisible(true);
    }
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
    public static void main(String[] args) {
        AWExample aweObj = new AWExample();
    }
}

```



Q2. Create a button and add a action listener for mouse click.

```

import java.awt.*;
import java.awt.event.*;

```

Public class EventHandling
ActionListener {

26

Frame f;

TextField tf;

EventHandling() {

//create components

f = new Frame();

f.addActionListener(this);

tf = new TextField();

tf.setBounds(60, 50, 170, 20);

Button b = new Button("click me");

b.setBounds(100, 120, 80, 30);

f.add(tf); f.add(b);

//register listener

b.addActionListener(this); //passing current instance.

b.setVisible(true);

//add components and set size, layout and visibility

f.add(b); f.add(tf);

f.setSize(300, 300);

f.setLayout(null);

f.setVisible(true);

}

public void actionPerformed(ActionEvent e) {

tf.setText("welcome");

}

public void windowClosing(WindowEvent e) {

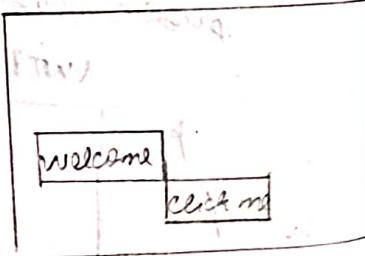
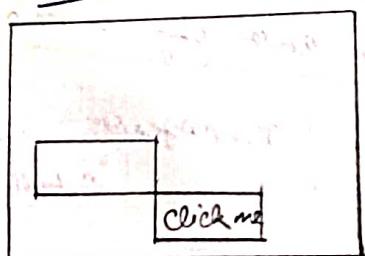
System.exit(0);

}

public static void main(String args[]) {

new EventHandling();

OUTPUT



PROGRAMS ON IO

1. Example 1

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

Q7

```

byte[] buf = {35, 36, 37, 38};  

// Create the new byte array input stream.  

ByteArrayInputStream bye = new ByteArrayInputStream(buf);  

int k = 0;  

while ((k = bye.read()) != -1) {  

    // conversion of a byte into character  

    char ch = (char) k;  

    System.out.println("ASCII value of character " + ch);  

    k + 1; // special character is "+ch";  

}

```

3

2. Example 2

```

import java.io.*;  

public class ByteArrayInput {  

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

        byte[] buf = {35, 36, 37, 38};  

    }

```

Hindi

```

// Create the new byte array input stream  

ByteArrayInputStream bye = new ByteArrayInputStream(buf);  

int k = 0;  

while ((k = bye.read()) != -1) {  

    // conversion of a byte into character  

    char ch = (char) k;  

    System.out.println("ASCII value of character " + ch);  

    k + 1; // special character is "+ch";  

}

```

Hindi

```

}

```

Example 3

```

public class FileEx {  

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

        BufferedReader fin = new FileInputStream ("Example.txt");  

        the content;  

        System.out.println ("Remaining bytes that can be read: " + fin.read());
    }
}

```

(28)

```

fin.available());
content = fin.read();
System.out.print((char) content + " ");
System.out.print(content + " ");
System.out.println("Remaining bytes that can be
read : " + fin.available());
System.out.println("Remaining bytes that can be read
: " + fin.available());
}

```

Example 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");
        byte[] bytes = new byte[20];
        int i;
        char c;
        i = fin.read(bytes);
        System.out.println("Number of bytes read : " + i);
        System.out.println("Bytes read : " );
        //for each byte in buffer
        for (byte b : bytes) {
            //converts byte to character
            c = (char) b;
            //print
            System.out.print(c);
        }
    }
}

```