

22-12-23

## LAB-I

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

⇒ class Overload {

    void point (int n) {

        int sum = 0;

        for (int i=1; i <=n; i++) {

            sum += i;

        System.out.println ("Sum of " + n + " natural  
nos is " + sum);

}

    void point (int m, int n) {

        System.out.println ("Prime nos. 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);

3

3

3

class overload Demo {

public static void main (String[] args) {

overload o = new Overload();

o.print(5);

o.print(13);

3

off :- Sum off 5 natural no. i.e 15

pair no. no. in the range 0-15

7

11

13

class Overload {

String c-name;

String c-ph;

double total;

Overload (String cname, String c-ph) {

this.c-name = cname;

this.c-ph = c-ph;

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

total = q-dal \* 100 + q-pulses \* 80 + q-sugarcane \* 50;

void display ()

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

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

3

3

60

class Order{}

public static void main(String[] args){

Grocery g1 = new Grocery('Rama', '8060430');

Grocery g2 = new Grocery('Sham', '76846325');

Grocery g3 = new Grocery('Bhavana', '96328874');

g1.calc(2, 2, 1);

g1.display();

g2.calc(3, 5, 2);

g2.display();

g3.calc(1, 1, 0.5);

g3.display();

3

104

Name Phone-No Total

Rama 8060430 410.00

Name

Phone-No

Total

Chama 76846325 180.00

Name

Phone-no

Total

Bhavana 96328874 205.00

Ques - Write a program to calculate the total amount of a grocery bill.

`import java.util.Scanner;`

`class QuadS`

`int a, b, c;`

`double root1, root2, d;`

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

`void input() {`

`System.out.println("Quadratic eq' de in the  
format : 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");`

`root1 = (-b + Math.sqrt(d)) / (2*a);`

`root2 = (-b - Math.sqrt(d)) / (2*a);`

`System.out.print("First root is: ", root1);`

`System.out.print("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);

3

else {

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

$$\text{double real} = -b / (2 * a)$$

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

System.out.println ("This eqn has two complex roots: " + real + "+" + imaginary + "i" + " " + real + "-" + imaginary + "i");

3. Output's returning to main.

3

Method returning to main.

(A return statement)

Class MainSystem

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

Quad q = new Quad ();

q.input ();

q.discriminant ();

q.calculateRoots ();

3. Output's returning to main.

3

(Q) Enter a: 1

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

Enter b: 2

Enter c: 2

Roots are real & unequal.

First root is: 0.0

Second root is: -2.0

$(x^2 - 4x - 4) = 0$

27-24

(1)

want a Java program to create a class Student with members "IDN", name, marks (6 subjects). Include methods to accept student details & marks. Also include a method to calculate the percentage & display "appropriate" details (Mention subject to be cleared).

$\Rightarrow$  import java.util.Scanner;

class Student {

String IDN;

String name;

double[] marks = new double[6];

void inputDetails() {

Scanner scanner = new Scanner(System.in);

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

IDN = scanner.nextInt();

System.out.print("Enter 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();

System.out.println();

i++;

double calculatePercentage() {

double totalmarks = 0.0;

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

totalmarks += marks[i]; }

return (totalmarks / 6);

sp

void displayDetails() {  
System.out.println("Student Details");  
System.out.println("USN: " + USN);  
System.out.println("Name: " + name);  
System.out.println("Marks: " + marks);  
for (int i = 0; i < 6; i++) {  
System.out.println("Subject " + (i + 1) + ": " +  
marks[i] + " Grade: " + grade[i]);  
}  
}

System.out.println("Percentage: " + calculatePercentage());

class Student {  
String name;  
int marks[];  
double percentage;

public static void main(String[] args) {  
Scanner scanner = new Scanner(System.in);  
System.out.println("Enter no. of students: ");  
int numberofStudents = scanner.nextInt();  
Student student[] = new Student[numberofStudents];  
for (int i = 0; i < numberofStudents; i++) {  
student[i] = new Student();  
}

for (int i = 0; i < numberofStudents; i++) {  
System.out.println("Enter details for  
Student " + (i + 1) + ":");  
student[i].inputDetails();  
}

for (int i = 0; i < numberofStudents; i++) {  
student[i].displayDetails();  
}

Jo  
J  
J

Op: Enter details for student 1 : aa, usn: abm2

Enter USN: abm2

Enter name: aa

Enter marks for 6 subjects

Subject 1: 90

Subject 2: 92

Subject 3: 93

Subject 4: 87

Subject 5: 88

Subject 6: 98

Enter details for student 2 : bb, name: bb, usn: abm2

Enter USN: abm2

Enter name: bb

Enter marks for 6 subjects: 90, 92, 93, 87, 88, 98

Subject 1: 99

Subject 2: 97

Subject 3: 100

Subject 4: 79

Subject 5: 94

Subject 6: 89

### Student Details:

USN: abm2

Name: aa

Marks: Subject 1: 90, 91, 92, 93, 94, 95

Subject 2: 92

Subject 3: 93

Subject 4: 87

Subject 5: 88

Subject 6: 98

percentage: 91.33333333%

Student details: Name: [redacted] Address: [redacted] SP: [redacted]  
 USN: 1bm2 Number: [redacted] Grade: [redacted] Marks: [redacted]  
 Maths: Subject 1: 99  
 Subject 2: 97  
 Subject 3: 100  
 Subject 4: 79  
 Subject 5: 94  
 Subject 6: 83  
 Percentage: 93.01

(2) Create a class Book that contains 4 data members: name, author, price, & numPages. Include a constructor to set the values for the members. Include methods to set & 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.

→ implement java.util.Scanner;

class Book{

String name;

String author;

int price;

int numPages;

Book(String name, String author, int price, int numPages)

{ this.name = name;

this.author = author;

this.price = price;

this.numPages = numPages;

}

L-E-S-S-E-S - ID: [redacted]

```

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; String name; String author; int price;
    int numPages;
    System.out.println("Enter the no. of books");
    n = s.nextInt();
    Book b[] = new Book[n];
    for (int i=0; i<n; i++) {
        System.out.print("book" + (i+1));
        System.out.print("The name of the book:");
        name = s.next();
        System.out.print("Enter author:");
        author = s.next();
        System.out.print("Enter price:");
        price = s.nextInt();
        System.out.print("Enter no. of pages:");
        numPages = s.nextInt();
        b[i] = new Book(name, author, price, numPages);
    }
    for (int i=0; i<n; i++) {
        System.out.print("book" + (i+1) + " " + b[i]);
    }
}

```

}

}

→ Enter the name of book in "book name" field  
2 Enter the name of another book in "book name" field  
Enter author's name in "author name" field  
Enter publisher's name in "publisher" field  
Enter price:  
120

Enter no. of pages: 200

book 2:

Enter the name of book: XY, author name: ABCD

Enter author: fghij

Enter price: 150, number of pages: 300

book 1:

Book name: XYZ, author name: ABCD

Author name: ABCD

price: 120

number of pages: 200

book 2: Book name: XYZ, author name: ABCD

Book name: XYZ, author name: ABCD

Author name: fghij

price: 150, number of pages: 300

Number of Pages: 300, price: 150

total price: 300 \* 150 = 45000

(1) Develop a Java program to create an abstract class named "Shape" that contains two integers. It an empty method named "paintArea()". Provide three 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 paintArea() that prints the area of the given shape.

→ Abstract class Shape:

int side1;

int side2; // width & height dimensions

• class & field os: int b, int h, int area;

// Shape (int side1, int side2) { }

: (void) this.side1 = side1;

this.side2 = side2;

}

Abstract void paintArea();

3

Class Rectangle extends Shape { int b, int h; }

{ }

Abstract Super (b, h);

{ }

void paintArea() { }

int area = side1 \* side2; }

System.out.println("Area of rectangle is: " + area)

3

Class Triangle extends Shape { }

{ }

Abstract Super (b, h);

void printArea() {

int area = pi \* side1 \* side2; // calculate area  
 System.out.println("The area of the triangle  
 is " + area);  
 } // end of method printArea  
 } // end of class Triangle

Circle (int x, int y) {

super(x, y); // call constructor

radius = 5; // set radius

void printArea() {

double pi = 3.14 \* radius \* radius;

System.out.println("The radius of the circle  
 is " + radius + " and area is " + area);

3 : 5.000 - 78.500

class Main {

public static void main(String[] args) {

Rectangle r = new Rectangle(2, 3);

Triangle t = new Triangle(2, 3);

Circle c = new Circle(7);

r.printArea(); // prints 6.000

t.printArea(); // prints 6.000

c.printArea(); // prints 153.933

3 : 6.000 - 6.000 - 153.933

3 : 6.000 - 6.000 - 153.933

Area of rectangle is 6.000

Area of Circle is 153.933

Date \_\_\_\_\_  
Page \_\_\_\_\_

(II) Develop a Java program to create a "SBI BANK" that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facility but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance "level" of the balance falls below this level, a service charge is imposed. Create a class account that stores customer name, account number & type of account. From this, derive the class Current & Savings to make them specific to their requirements. Include the necessary methods in order to achieve the following steps:

- (a) Accept deposit from customer and update the balance.
  - (b) Display the balance.
  - (c) Compute & deposit interest.
  - (d) Permit withdrawal or update the balance:
    - check for minimum balance, impose penalty if necessary & update the balance.
- ⇒ import java.util.Scanner;

class Account { } (class definition)

String name;

int auro;

String type; } (class definition)

double balance; }

Account (String name, int auro, String type, double balance)

this.name = name; this.auro = auro;

if (this.auro <= 0) { throw new Error(); }

this.type = type;

```

1. withdraw = balance - amount;
2. void deposit (double amount) {
    balance += amount;
}
3. void withdraw (double amount) {
    if ((balance - amount) >= 0) {
        balance -= amount;
    }
}
4. System.out.println ("name: " + name + " account no: " + accountNo + " balance: " + balance);

```

Class Savings extends Account

```

private static double rate = 5;
Savings (String name, int accNo, double balance) {
    super (name, accNo, "savings", balance);
}
void interest () {
    balance += balance * rate / 100;
}
System.out.println ("balance: " + balance);
}

```

Class Current extends Account

```

private double minBal = 500;
private double serviceCharge = 5;

```

```

Current (String name, int accNo, double balance) {
}

```

```

void checkMin () {
    if (balance < minBal) {
}

```

```

System.out.println ("balance less than min
balance, service charge
")

```

```

balance -= serviceCharge;
}

```

```

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

```

class accountmain {

    public static void main (String args) {

        Scanner s = new Scanner (System.in);

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

        String name = s.nextLine();

        System.out.println ("Enter the type - current savings ");

        String type = s.nextLine();

        System.out.println ("Enter the account no: ");

        int accno = s.nextInt();

        System.out.println ("Enter the initial balance: ");

        double balance = s.nextDouble();

        int ch;

        double amount1, amount2;

        Account ac = new Account (name, accno, type, balance);

        SavingsAcct sa = new SavingsAcct (name, accno, balance);

        CurrentAcct ca = new CurrentAcct (name, accno, balance);

        while (true) {

            if (accType.equals ("savings")) {

                System.out.println ("1) deposit 2) withdraw

                3) compute interest 4) display ");

                System.out.println ("Enter the choice: ");

                ch = s.nextInt();

                switch (ch) {

                    case 1: System.out.println ("Enter the amount ");

                        amount1 = s.nextInt();

                        sa.deposit (amount1);

                        break;

                    case 2: System.out.println ("Enter the amount ");

                        amount2 = s.nextInt();

                        sa.withdraw (amount2);

                        break;

case 3: `ca::interest()`

`break;`

; case 4: `ca::display()`

`break;`

; case 5: `System.exit(0)`

default: `System.out.println("invalid input")`

`break;`

`else if`

`System.out.println("choose 1 for deposit, 2 withdrawl,`

`3 display");`

`System.out.println("Enter the choice:");`

`ch = s.nextInt();`

`switch(ch){`

case 1: `System.out.println("Enter the amount");`

`amount_1 = s.nextInt();`

`ca::deposit(amount_1);`

`break; else if`

case 2: `System.out.println("Enter the amount");`

`amount_2 = s.nextInt();`

`ca::withdrawl(amount_2);`

`ca::checkmin();`

`break; else if`

case 3: `ca::display();`

`break;`

case 4: `System.exit(0);`

default: `System.out.println("invalid input");`

`break;`

`J J J`

`if (choice == 1) {`

`if (amount >= 5000)`

`if (amount <= 10000)`

`amount`

16-2-24

## Exception Handling

(Q) write a program that demonstrates handling of exceptions in inheritance. Create a base class called 'father' & a derived class called 'son', which extends the base class. In father class, implement a constructor, which takes the age & therefore the exception will be thrown when the given age < 0. In son class, implement a constructor that calls both father's constructor & therefore throws exception if son's age is  $\geq$  father's age.

⇒ import *java.awt.Scanner*

public class Exception implements [most of]

static class Father { }

600 6 int page;

Father (int page) therefore writing age?

~~If (age < 0) {~~

wherever new message goes ("catchall")  
they will return and establish age can't be negative! ) ;  
is enough to do this: ~~define~~ = function ( ) {

~~so early that it was this forge = forge, writing~~

student at first stage

int main() {  
 int gauge, target;  
 cout << "Enter current gauge reading: ";  
 cin >> gauge;  
 cout << "Enter target gauge reading: ";  
 cin >> target;  
 if (gauge < target) {  
 cout << "The current gauge reading is less than the target." << endl;  
 } else if (gauge == target) {  
 cout << "The current gauge reading is equal to the target." << endl;  
 } else {  
 cout << "The current gauge reading is greater than the target." << endl;  
 }  
}

*Scutellaria* (page 1) is equal to *Scutellaria*

if sage > fage { do something }

Every book sale I do I try to teach them new literary age (yours age  
isn't it?) so when I tell them can't be copied or greater than

(*father's language*);  $\{\}$

4.1.34) ~~24 hours~~ HNS. sage = sage; ~~11:00~~ in ~~morning~~

J J . It will take 6000 - 6500 cubic feet.

static class Warnings() extends exception{}

Wavy edge (Creasy message) • f wavy  
edge (message)

suffix (mess age))

三

public static void main (String args[])
 {
 Scanner s = new Scanner (System.in);
 System.out.println ("Enter father's age:");
 int fatherAge = s.nextInt();
 System.out.println ("Enter Son's age:");
 int sonAge = s.nextInt();
 if (sonAge < fatherAge)
 {
 System.out.println ("Son is younger than father");
 }
 else if (sonAge == fatherAge)
 {
 System.out.println ("Son is equal to father");
 }
 else
 {
 System.out.println ("Son is older than father");
 }
 }
}

Q3. Create a package: CTF which has two classes Student and Internal. The class Student has members like roll, name, etc. The class Internal contains information about marks scored in five courses of the current semester of the student. Create another package: SFE which has the class Internal which is a derived class of Student; this class has an array that stores the SFE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

- (1) Create a folder, CTF where the programme Student.java is inside java within it.
- (2) Create a folder, SFE in B and the programme Internal.java within it.

- (3) Save the main program outside the class folder  
 (4) Compile Main.java & Execute main method.

→ package SFE;

import java.util.Scanner;

import java.util.Scanner;

public class SFE extends implements

protected int marks[];

protected final int max;

public void calculate()

{ final int marks = new int[5]; }

final int marks = new int[5];

public void SFE marks() { }

Scanner scanner = new Scanner(System.in);

System.out.println("Enter SFE marks for:");

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

System.out.println("Subject " + (i+1) + " mark  
marks[i] = scanner.nextInt();

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

public void calculateFinalMarks()

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

finalMarks[i] = marks[i] / 2 + superMarks[i];

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

public void displayFinalMarks()

System.out.println("Subject " + (i+1) + " : " +

System.out.println("Subject " + (i+1) + " : " +

System.out.println("Subject " + (i+1) + " : " +

System.out.println("Subject " + (i+1) + " : " +

System.out.println("Subject " + (i+1) + " : " +

```

package ctf;
import java.util.Scanner;
public class Internate extends Student {
    protected int marks[] = new int[5];
    public void Input() {
        Scanner Scanner = new Scanner(Systems.in);
        Systems.out.println("Enter internal marks from three subjects");
        for (int i=0; i<5; i++) {
            Systems.out.print("Subject " + (i+1) + " marks: ");
            marks[i] = Scanner.nextInt();
        }
    }
    public void Output() {
        for (int i=0; i<5; i++) {
            Systems.out.print(marks[i] + " ");
        }
    }
}

import java.util.Scanner;
public class Main {
    public static void main (String args[]) {
        int numofStudents = 2;
        Extended finalMarks[] = new Extended [numofStudents];
        for (int i=0; i<numofStudents; i++) {
            finalMarks[i] = new Extended();
            finalMarks[i].Input();
            finalMarks[i].Output();
        }
        Systems.out.println("Enter CSE marks");
        finalMarks[0].Input();
        Systems.out.println("Enter CSE marks");
        finalMarks[1].Input();
        Systems.out.println("Inputting data " + n);
        for (int i=0; i<numofStudents; i++) {
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].displayFinalMarks();
        }
    }
}

```

```

package CIE;
import java.util.Scanner;
public class Student {
    protected String USN;
    protected String name;
    protected int sem;
    public void inputStudentDetails() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter USN: ");
        USN = scanner.next();
        System.out.print("Enter name: ");
        name = scanner.next();
        System.out.print("Enter Semester: ");
        sem = scanner.nextInt();
    }
    public void displayStudentDetails() {
        System.out.println("USN: " + USN);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}

```

(3) → Write a program which creates two threads, one thread displaying "BMS college of Engineering" once every ten seconds & another displaying "CSE" once every two second.

→ class displayThread extends Thread {

```

private String message;
private int interval;
public displayThread(String message, int interval) {
    this.message = message;
    this.interval = interval;
}

```

public void run() {  
try {

  while (true) {

    System.out.println("message");

    Thread.sleep(5000);

} // try

} // catch (InterruptedException e)

  e.printStackTrace();

} // catch (Exception e)

public class Main {  
  public static void main (String [] args) {

    DisplayThread thread1 = new DisplayThread

( "BMS college of Engineering", 10000);

    DisplayThread thread2 = new DisplayThread

( "BMS college of Engineering", 2000);

    thread1.start();

    thread2.start();

}

}, public class Main {  
  public static void main (String [] args) {

    DisplayThread thread1 = new DisplayThread

( "BMS college of Engineering", 10000);

CSE

    DisplayThread thread2 = new DisplayThread

( "BMS college of Engineering", 2000);

    thread1.start();

    thread2.start();

  }, public class Main {  
    public static void main (String [] args) {

      DisplayThread thread1 = new DisplayThread

( "BMS college of Engineering", 10000);

      DisplayThread thread2 = new DisplayThread

( "BMS college of Engineering", 2000);

      thread1.start();

      thread2.start();

CSE

CSE

CSE

## LABS

23-2-24

(1) Creating label, button & Text Field in a frame using AWT.

```
-> import java.awt.*;
import java.awt.event.*;
public class AWTExample extends WindowAdapter {
    Frame f;
    Label l;
    Button b;
    TextField t;
```

```
f = new Frame();
f.addWindowListener(this);
Label l = new Label("Employee ID:");
Button b = new JButton("Submit");
TextField t = new JTextField();
l.setBounds(20, 80, 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 obj = new AWTExample();}
```

JP(C) preparing window, when time is set, it will open  
JP(C) will print C when open

Q/P

employee input

1234

Submit button

rose field

(2)

Create a button & add a action Listener for Name click.

=> import javax.awt.\*;

import javax.awt.event.\*;

public class EventHandling extends WindowAdapter

implements ActionListener

EventHandling () {

f = new Frame();

f.addWindowListener (this);

tf = new TextField();

tf.setBounds (60, 50, 70, 220);

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

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

b.addActionListener (this);

f.add(b); f.add(tf); f.setSize (300, 300);

f.setLayout (null);

f.setVisible (true);

public void actionPerformed (ActionEvent e) {

tf.setText ("welcome");

public void windowClosing (WindowEvent e) {

System.exit (0);

public static void main (String args[]) {

new EventHandling ();

of

123

welcome

Welcome

click me

## (3) Programs on TIO.

`import java.io.*;`

`public class ByteArrayInput {`

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

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

`ByteArrayInputStream bfr = new ByteArrayInputStream (buf);`

`int k = 0;`

`while (k < buf.length) {`

`char ch = (char) bfr.read ();`

`System.out.println ("ASCII value of character is . " +`

`ch + " + " + (k + 1));` special character is: "

`k++;`

`}` joining them with `,`

`int k = 0;` joining them with `,`

`if (k == 3) {`

~~→ ASCII value of character is -1 : special character i.e. ?~~

(II) `import java.io.*;` file output stream writing :-

`public class ByteArray - er {`

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

`FileOutputStream fout = new FileOutputStream`

`("Example . txt");`

`FileOutputStream f1 = new FileOutputStream`

`("Example2 . txt");` writing below;

`ByteArrayOutputStream bout = new ByteArrayOutputStream`

`();` write to Stream();

`bout.write (5);` or `write (5)`

`bout.write (6);` or `write (6)`

`bout.write (7);` or `write (7)`

`bout.close ();`

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

off

current...

(11) + import java.io.\*;  
public class fileEx {  
public static void main (String args[]) throws  
IOException {  
FileInputStream fin = new FileInputStream ("Example.txt");  
int content; // int content; int content; int content;  
System.out.println ("Remaining bytes that can be  
read: " + fin.available());  
while ((content = fin.read ()) != -1) {  
System.out.print ((char) content + " ");  
System.out.print (content + " ");  
System.out.println ("In Remaining bytes that can  
be read: " + fin.available());  
fin.close();  
}

⇒ Remaining bytes that can be read: 11  
A 65

remaining bytes that can be read: 0

-it. import java.io.FileInputStream;  
import java.io.IOException;  
public class fileEx {  
public static void main (String args[]) throws IOException {

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

byte [] bytes = new byte [20];

fin.read (bytes);

```
int i; char c;  
i = fin.readBytes();  
System.out.println("No. of bytes read: " + i);  
System.out.print("Bytes read: ");  
for (byte b : bytes) {  
    c = (char) b;  
    System.out.print(c);  
}  
}
```

~~Q:~~  
~~Number of bytes read: 1~~  
~~Bytes read: P~~

~~Q~~  
~~212124~~