

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

I/P → class overload fns. Lecture 2. File no. - 916

```
void point(int n){ // n. Natural numbers
    int sum = 0; // sum of natural numbers
    for (int i = 1; i <= n; i++) {
        sum = sum + i;
    }
}
```

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

else { // if two numbers are passed
 int m, n; // m & n
}

```
if (m < n) {
    void point(int m, int n){ // m & n
        System.out.println("Prime numbers in the range
            from " + m + " to " + n + " 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);
}
```

if (flag == 0)

System.out.println(i);

else { // if two numbers are passed
 int m, n; // m & n
}

if (m < n) {
 void point(int m, int n){ // m & n
 System.out.println("Prime numbers in the range
 from " + m + " to " + n + " are");
 }
}

Class overload Demo { }

Public static void main(String[] args) { }

function Overload O=new overload(); after. (1-0)
constructor Overload(5); In main having total taking
Overload(7,13); constructor sum number
 3 types of overload o w/ load more number and
 by adding sum performance is better.

O/P - Sum of 5 natural numbers is 15 or 19 (Q-1)
 prime numbers in the range 1 to 100

7

10 = max tri

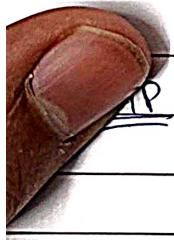
11

{ (7+8+9+10+11=50) / 5 = 10 tri } not

13

14 = max even

(Q-2) Write a Java 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. The method to return the total price. Display the name, ph_no and total bill of 3 customers.



```

  class grocery {
    String c_name;
    String c_Phone;
    double total;

    grocery(String c_name, String c_Phone) {
      this.c_name = c_name;
      this.c_Phone = c_Phone;
    }

    void calc(double q_dal, double q_pulses,
             double q_sugar) {
      total = q_dal * 100 + q_pulses * 80 + q_sugar * 50;
    }

    void print() {
      System.out.println("Name: " + c_name);
      System.out.println("Phone: " + c_Phone);
      System.out.println("Total: " + total);
    }
  }

```

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

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

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

```
3
```

```
class Demo {
```

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

```
Biology g1 = new Biology("Rama", "8060302010");
```

```
Biology g2 = new Biology("Shama", "7689632510");
```

```
Biology g3 = new Biology("Rama", "9632587412");
```

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

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

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

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

```
g3.calc(1, 0, 5);
```

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

```
3
```

```
3
```

O/P: Name Phone number Total

Rama 8060302010 410.00

Shama 7689632510 800.00

Rama 9632587412 205.00

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

Q18) `import java.util.Scanner;`

`class Quadratic`

`{` `public void main()`

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

`double a, b, c, d;`
`s. put 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");`

~~if (2 roots - discriminant) then~~ ~~if (one root - discriminant)~~

~~else (no roots - discriminant < 0)~~ ~~else (no roots - discriminant < 0)~~

`else if (discriminant == 0) {`

`System.out.println("Enter a:");`

`a = scanner.nextInt();`

`System.out.println("Enter b:");`

`b = scanner.nextInt();`

`c = scanner.nextInt();`

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

`if (d >= 0) {`

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

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

`System.out.println("First root is " + root1);`

`System.out.println("Second root is " + root2);`

`else {`

`System.out.println("Roots are real and equal");`

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

`System.out.println("Root : " + root1);`

`} else {`

`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 : " + root1);`

`System.out.println("Root2 : " + root2);`

`} else {`

`System.out.println("Roots are imaginary");`

`} else {`

`System.out.println("Roots are complex");`

System out of problem ("no real roots") - Roots
 are complex numbers with "one imaginary");
 \Rightarrow double root = $b/(2 \cdot a)$;
 a double imaginary = $i \sqrt{-(-\Delta)/(2 \cdot a)}$;
 b) if system out of problem like the equation has two
 complex roots: "real + " + imaginary + "i and"
 "real + " - " + imaginary + "i";

{ 3. 2020. 01. 18. 10. 00. 02: Friday 9:12

{ Space Standard 7 working mode

{ 2020. 01. 18. 10. 00. 02: Friday 9:12

class main {

public static void main (String[] args) {

quad q = new quad();

q . input();

(new q). discriminant();

(new q). calculateRoots();

if ((new q). discriminant() < 0) {

{ (new q). discriminant();

(new q). discriminant();

O/P Quadratic equation in the form: $a x^2 + b x + c$

Enter a: 5 / (1 + P . d * 9 + Q * d * P) and

Enter "abs.2." into q . discriminant();

Enter "1.0" (bottom, 1) = 1.0

no real solutions. Roots are imaginary. The
 equation has two complex roots: $0.0 + 0.7483 i$
 $14.773547882 i$ and $0.0 - 0.7483 i$.

{ (new q). discriminant(); and

q . calculateRoots();

(a) Direct Differentiation

{ 2020. 01. 18. 10. 00. 02: Friday 9:12

{ (b) Partial Derivative Method

LAB-2

Q-1) Write 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, find percentage and display appropriate details.

IP import java.util.Scanner;

class Student {

 String USN;

 String name;

 double[] marks = new double[6];

 }

 void acceptDetails() {

 Scanner sc = new Scanner(System.in);

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

 USN = sc.nextLine();

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

 name = sc.nextLine();

 System.out.print("Enter marks for 6 subjects");

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

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

 marks[p] = sc.nextDouble();

 }

 double totalMarks = 0;

 for (double mark : marks) {

 totalMarks += mark;

 }

 return (totalMarks / 6);

}

void displayDetails() {

 System.out.println("Student Details");

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

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

System.out.println ("marks: ");

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

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

marks[i]);

}

public class Student {

public static void main (String [] args) {

Scanner sc = new Scanner (System.in);

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

int numStudents = sc.nextInt ();

Student [] students = new Student [numStudents];

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

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

student[i] = new Student ();

student[i].acceptDetails ();

student[i].displayDetails ();

System.out.println ("Student " + i + " details accepted");

System.out.println ("Student " + i + " details displayed");

System.out.println ("Press any key to quit");

System.out.println ("Press any key to quit");

System.out.println ("Press any key to quit");

OP Enter no. of students: 2

Enter details for student 1:

Enter USN: 18M22CS361

Enter name: Mohan

Enter marks for 6 subjects:

Subs 1: 50 Subs 2: 60 Subs 3: 70 Subs 4: 80 Subs 5: 90 Subs 6: 100

Subs 1: 50 Subs 2: 60 Subs 3: 70 Subs 4: 80 Subs 5: 90 Subs 6: 100

Subs 1: 50 Subs 2: 60 Subs 3: 70 Subs 4: 80 Subs 5: 90 Subs 6: 100

Subj 1: 90% (including assignments)

Subj 2: 80% (including assignments)

Subj 3: 60% (including assignments)

Subj 4: 70% (including assignments)

Marks: student details: Name: Ishaan Patel

UIN: 2018PM22S361

Name: Ishaan Patel

Marks:

Subj 1: 200 out of 200 marks = 100%

Subj 2: 180 out of 200 marks = 90%

Subj 3: 90 out of 100 marks = 90%

Subj 4: 50 out of 100 marks = 50%

Subj 5: 160 out of 200 marks = 80%

Subj 6: 160 out of 200 marks = 80%

Percentage: 40.33% (including assignments)

(Create) Book class Book must have 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 book. Develop a Java program to create

'n' book objects.

IIP

import java.util.Scanner;

class Books {

Scanner s = new Scanner (System.in);

String name, author, price; int numPages;

Books (String name, String author, int price, int

numPages) {

this.name = name;

this. author = author;

this. price = price;

this. numpages = numpages;

public string tostring() { return name + " " + author + " " + price + " " + numpages; }

public string tostring() { return name + " " + author + " " + price + " " + numpages; }

String bookName = "Book Name" + this.name
+ " " + "in";

String authName = "Author Name" + this.author;

String bookPrice = "Price" + this.price;

return bookName + authName + bookPrice + pages;

}

}

class BookProgram {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int n;

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

n = s.nextInt();

Books b[] = new Books[n];

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

System.out.println("Details of books:" + (i + 1) +
{});

System.out.println("Enter name of Book");

String name = s.next();

System.out.println("Enter name of Author");

String author = s.next();

int price = s.nextInt();

System.out.println("Enter no. of pages");

int numpages = s.nextInt();

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

}

for (Books book : b) {

System.out.println(book.tostring());

}

}

OP Enter number of book: 1000 - Part

BOOK 1000 - Page = 500 - Line

Enter name of book: Jungle Book

Enter author of book: Rudyard Kipling

Enter price of book: 1000 - 00 per set - 1000

Amount: 500 - wood price

Amount other: 0

Amount of Book 1000 - amount printed

Book Name: Jungle Book - wood printed

Author: Rudyard Kipling - Wood Printed

price: 1000

pages: 500

Amount of Book 1000 - wood printed

CD demand to wood price

Q-1 Develop a C++ program to create an abstract class named "Shape" that contain two integers and an empty method named `paintArea()`. Provide three classes named Rectangle, Triangle and Circle such that each one of these classes extends this class Shape. Only ob the classes contain the method `paintArea()` that prints the area of the given shape.

I/P: abstract class Shape

protected int side1;

protected int side2;

public Shape (int side1, int side2) {

 this.side1 = side1; this.side2 = side2;

 }

 public void paintArea();

 }

Class Rectangle extends Shape

public Rectangle (int length, int width) {

 super(length, width);

}

public void paintArea() {

 int area = side1 * side2;

 System.out.println("Area of Rectangle: " + area);

}

Class Triangle extends Shape

public Triangle (int base, int height) {

 super(base, height);

}

public void paintArea() {

Inside two side double area = $0.5 \times \text{side}_1 \times \text{side}_2$; i.e. $\frac{1}{2} \times \text{base} \times \text{height}$
 system.out.println("Area of Triangle: " + area);
 triangle extends Shape class
 public class Shape {
 public void printArea() {
 System.out.println("Super Class");
 System.out.println("Area of circle: " + pi * radius * radius);
 System.out.println("Area of rectangle: " + length * width);
 System.out.println("Area of triangle: " + 0.5 * base * height);
 }
 }

Public class program 12: $= 1 \times 2 \times 2 \times \pi$

public static void main (String[] args) {

 Rectangle rectangle = new Rectangle(4, 6);

 Triangle triangle = new Triangle(4, 6);

 Circle circle = new Circle(4);

 rectangle.printArea();

 triangle.printArea();
 circle.printArea();

Output in console-printArea() method printing

3. (Ans) Area of rectangle = $4 \times 6 = 24$

3. (Ans) Area of triangle = $\frac{1}{2} \times 4 \times 6 = 12$

3. (Ans) Area of circle = $\pi \times 4^2 = 50.265482457435$

O/P Area of rectangle: 24.000000000000004

O/P Area of triangle: 12.000000000000002

O/P Area of circle: 50.265482457435

Q-2 Develop a Java program to generate a bank account
 bank account management: two kinds of account
 for its customers, one called savings account
 and the other current account. The savings
 account provides compound interest and

withdrawal facilities but no negative book facility.

The current account provides cheque book facility but no interest. Current account holders would also maintain a min balance and if the balance falls below this level, a service charge will be imposed & credit a user.

Account has customer name and type of account. In this derive the class current and saving acc & make them more specific to user requirements.

(a) Accept deposit from customer & update the balance.

(b) Display the balance.

(c) Compute demand deposit interest.

(d) permit withdrawal and update the balance. If the min balance is less than zero, impose penalty at necessary and update the balance.

I/P import java.util.Scanner;

class Account {

protected String customerName;

protected long accountNumber;

protected String accountType;

protected double balance;

public Account (String customerName, long

accountNumber, String accountType,

double balance) {

this.customerName = customerName;

this.accountNumber = accountNumber;

this.accountType = accountType;

this.balance = balance; }

from public void displayBalance() { }

System.out.println("Account Number: " + accountNumber);

System.out.println("Customer name: " + customerName);
System.out.println("Account Type: " + accountType);
System.out.println("Balance: " + balance);
System.out.println("Deposit: " + amount);
balance += amount;
System.out.println("Deposit of \$" + amount + " successful.");
displayBalance();

Public void withdraw(double amount){
if (amount <= balance){
balance -= amount;
System.out.println("Insufficient funds");
}
displayBalance();

Class Current extends Account{
private double minBalance = 1000;
private double serviceCharge = 50;
Public Current(String customerName, long accountNumber, double balance){
super(customerName, accountNumber, "Current", balance);
}
Public void withdraw(double amount){
if (amount <= balance - minBalance){
balance -= amount;
System.out.println("Balance: " + balance);
}
}

System of old printing ("window of \$") + amount

(After 3 days of work) : It's "successful"

3. Concurrenza tra i diversi
recessi

~~System.out.println("Invisible Render Scrape")~~

• 6028

charge of "F" + service charge.

(Inventor, Assignee - If simplified)

display(Balance); Sat : 2021-07-01 10:00:00

3

class Contact extends Account {

private double interestRate = 0.05;

public void SetCustomerName(string customerName) { }

Long Account Number:

doubtre balance) $E_{\text{min}}(x)$

Super customer name, account number,

"Scalpings", bald (am 19); 39-9

3 220789 2000-07-15

void computePixelIndex();

double interest = balance * interest rate

balance e + 2 interessi

display Balance (0) 394.87 1000000

Deborah Bonnicksen

Digitized by srujanika@gmail.com

~~Public class programming~~

Public static void main (String[] args) {

~~scanner sc = new Scanner (System.in);~~

~~Caravat saving Act = new Savillet ("Jann Doe")~~

12345, 5000);

Saving Ac. Display Balance();

business: 8. In saving Accts deposit (1000);
• (100) saving acct. computation etc();
• saving acct. withdrawal (2000);
{ 9/12.

Business: current account (current accts "John Doe", 987654321);
withdrawal: 8. for spending money (1500);
• (150) current accts. display balance();
current accts. deposit (500);
current accts. withdrawal (500);
sc. date();
8. into withdrawal (dinner amount);
8. (cash withdrawal withdrawal amount);
\$20.00 = 86. After withdrawal balance = 3601.00

O/P Account Number: 123456
Customer Name: John Doe

Account Type: savings

Balance: \$6000.00

Deposit of \$1000 credited

Account number: 123456

Customer name: John Doe

Interest of \$300.00 credited. final balance

Account number: 987654321

Customer name: John Doe

Account type: current savings

Balance: \$2000.00

Insufficient funds

Account number: 987654

Customer name: John Doe

Balance: \$1950.00

\$100.00 withdrawn

\$100.00 withdrawn

\$100.00 withdrawn

Original balance \$2000.00

Q-1 Create a package CIE which has two classes Student and External. The class Student has members like USN, Name, Sem, Two arrays intervals described from student has an array that stores the internal marks scored in five courses of the student and another array that stores the marks scored in five courses of the student.

Create another package CIE which has the class External which is a derived class of Student. This class has a member array that stores the SEF marks scored in five courses of the student. Suppose one has packages in a file that declares the final marks of "n" students in all five courses.

IIP Package CIE,

import java.util.Scanner;

public class Student {

protected String USN = new String();

protected String name = new String();

protected int sem;

public void InputStudentDetails() {

Scanner sc = new Scanner(System.in);

System.out.println("Give USN");

USN = sc.nextLine();

System.out.println("Give name");

name = sc.nextLine();

System.out.println("Give Sem");

Sem = sc.nextInt();

public void displayStudentDetails() {

System.out.println("The USN is : " + USN);

System.out.println("The name is : " + name);

methods can be System.out.println("The sum is " + sum);
 protected int calculateSum() {
 int sum = 0;
 for (int i = 0; i < numbers.length; i++) {
 sum += numbers[i];
 }
 return sum;
 }

```

import java.util.Scanner;
public void input() {
  Scanner scanner = new Scanner(System.in);
  for (int i = 0; i < 5; i++) {
    System.out.print("Enter marks for course " + (i + 1));
    int marks = scanner.nextInt();
    total += marks;
  }
}
  
```

~~Package SEF~~

~~import CIF.Interval;~~
~~import java.util.Scanner;~~
~~public class ExternalMarks extends Interval {~~

~~protected int marks[5] of borrowing~~

~~protected int finalMarks[5] of borrowing~~

~~final marks[5] = new int[5];~~

Package SEE;

public class Father extends Grandfather;

{ public Father(); } { constructor(); }

public class Main { }

public static void main (String args) { }

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

final marks[i] = new Father();

final marks[i] = new Father();

final marks[i].input(i); marks[i];

System.out.println("Display data");

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

final marks[i] = calcFinalMarks();

"Display data" and "New marks array";

3 calculate final marks by adding all marks;

System.out.println("Final marks");

Q2 write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and the derived class called "Son" which extends the base class. In Father class implement a constructor which takes age and throw the exception WrongAge() when the input age < 0. In Son class implement a constructor that takes both Father and Son's age and throws an exception if Son's age is greater than Father's age.

"Display Data";

I/P import java.util.Scanner;

class WrongAge extends Exception { }

public WrongAge() { }

super ("message: Age can't be negative");

3 constructor (Scanner sc)

public WrongAge(String message) { }

super (message);

3

class InputScanner {

public int readInt() {

try (Scanner sc = new Scanner(System.in)) {

return sc.nextInt();

} catch (Exception e) {

System.out.println("Please enter a valid integer");

} else (Exception e) {

class Father extends InputScanner {

protected int fatherAge;

public Father() throws WrongAge {

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

fatherAge = readInt();

if (fatherAge < 0) {

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

public void display() {

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

class Son extends Father {

class Son extends Father {

protected int sonAge; // son's age

public Son() throws WrongAge {

super();

configureReadInt();

if (sonAge >= super.fatherAge) {

throw new WrongAge("Son's age cannot be

greater than father's age");

else if (sonAge < 0) {

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

public void display() {

super.display();

System.out.println("Son's age: " + sonAge);

```
public class Example {  
    public static void main(String[] args) {  
        try {  
            Son son = new Son();  
            son.display();  
        } catch (InterruptedException e) {  
            System.out.println("Error: " + e.getMessage());  
        }  
    }  
}
```

Q-3 Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSF" once every two seconds.

```
IP 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("CSF");  
                try {  
                    Thread.sleep(20000);  
                } catch (InterruptedException e) {  
                    e.printStackTrace();  
                }  
            }  
        };  
        thread1.start();  
        thread2.start();  
    }  
}
```

```
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("CSF");  
        try {  
            Thread.sleep(20000);  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
    }  
};  
thread1.start();  
thread2.start();
```

~~1. print stack tools (); 2. print preview~~

~~3. printing from the software directly without saving~~

~~4. printing from the software directly without saving~~

~~5. print management~~

~~read 1. stack (); 2. copying & pasting~~

~~read 2. stack (); 3. copy & paste method~~

~~(copying & pasting) advantage, time, cost~~

~~disadvantage, time, cost~~

~~example BMS college of engineering in sector E-0~~

~~CSF department 20 serials - 2019 purchased August~~

~~CSF department purchased from Lulu books~~

~~BMS college of engineering 2019 not room~~

~~CSF department 2019~~

~~CSF department purchased from Lulu books~~

~~RMF college of engineering 2019 - 2020~~

~~CSF department purchased from Lulu books~~

~~CSF department purchased from Lulu books~~

~~(copying & pasting) time, cost~~

~~not~~

~~CSF department purchased from Lulu books~~

~~CSF department purchased from Lulu books~~

~~(copying & pasting) time, cost~~

~~CSF department purchased from Lulu books~~

Q-1) Creating label, button and Textfield in a frame using AWT

AWT - Abstract Window Toolkit - Java's original window toolkit.

Abstract Window Toolkit - Java's original window toolkit.

Step 1) import java.awt.*; // Importing package

import java.awt.event.*; // Importing package

public class AWTExample extends WindowAdapter {
frame f; // Declaration of frame

AWT Example {

f = new frame(); // Declaration of frame

f.addwindowlistener(this); // Declaration of frame

label l = new label("Employee id:");

(("button") b = new button("Submit"));

Text field t = new textfield();

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

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

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

f.add(l); // Add label to frame

f.add(b); // Add button to frame

f.add(t); // Add textfield to frame

f.setSize(400, 300); // Set size of frame

f.setTitle("Employee info");

f.setLayout(null); // Set layout of frame

f.setVisible(true); // Set visibility of frame

}

public void windowClosing(WindowEvent e) {

System.exit(0); }

public static void main(String[] args) {

AWTExample awt = new AWTExample();

Q-2) Create a button and add a action listener for mouse click.

Java.awt
java.awt.event

I/P

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

public class EventHandling extends WindowAdapter
implements ActionListener {

Frame f; // frame to handle events

TextField tf; // input field that needs update

EventHandling() {

frame f; // frame to handle events

tf = new TextField(" ");

tf.setBounds(160, 50, 170, 20);

button b = new Button("username");

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

b.addActionListener(this);

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

f.setSize(300, 300);

f.setLayout(null);

f.setVisible(true);

frame f; // frame to handle events

tf.setText("welcome");

public void actionPerformed(ActionEvent e) {

if (e.getSource() == welcome) {

System.exit(0);

public static void main (String[] args) {

new EventHand();

} // main method

class EventHand extends WindowAdapter {

frame f; // frame to handle events

tf.setText(" ");

tf.requestFocus();

tf.addActionListener(this);

EXAMPLES ON I/O↳ File Input/OutputExample ->(1)

import java.io.*; // import all classes of package

public class ByteArrayInput { // main class

public static void main(String[] args) {

throws IOException {

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

int k = 0; // (for, do, while)

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

char ch = (char) k; // (char)

System.out.println("Value of character

" is " + ch + " & its ASCII value is " + ch);

} // (do while loop) note that character is " + ch);

// (do while loop) note that character is " + ch);

} // (do while loop) note that character is " + ch);

} // (do while loop) note that character is " + ch);

Example ->(2)

public class FileEx {

public static void main(String args) throws

IOException {

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

Content = fin.read();

System.out.println("Remaining bytes that can

be read is " + fin.available());

Content = fin.read();

System.out.print((char) Content + " ");

Content = fin.read();

System.out.println("Remaining bytes that can

be read is " + fin.available());

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

Example - ③

```
import java.io.FileInputStream; (OC- program)
import java.io.IOException;
public class file {
    public static void main(String[] args) throws IOException {
        IOException & UncaughtException
        FileInputStream fin = new FileInputStream("example.txt");
        byte[] bytes = new byte[10];
        fin.read(bytes);
        for (byte b : bytes) {
            c = (char) b;
            System.out.print(c);
        }
    }
}
```

Example - 4

~~import java.io.*; // imported on 2/2/98~~

~~public class ByteArray extends Object~~

~~throws IOException~~

~~{~~

~~public void main (String args) throws~~

~~IOException~~

~~{~~

~~fileOutputStream fout = new FileOutputStream~~

~~("example1.txt");~~

~~fout.write ("two two one one");~~

~~fout.close ();~~

~~fileOutputStream fout2 = new FileOutputStream ("example2.txt")~~

~~fout2.write ("one one two two three three");~~

~~fout2.close ();~~

~~ByteArrayOutputStream bout = new ByteArrayOutputStream ();~~

~~bout.write (65); bout.write ('A');~~

~~bout.writeTO (fout1);~~

~~fout1.close ();~~

~~bout.writeTO (fout2);~~

~~fout2.close ();~~

fout = flush();

fout = close();

System.out.println("Success");

}

}

O/P Fig ①:

ASCII of char: 25 special char #

ASCII of char: 36 special char \$

ASCII of char: 37 special char %

ASCII of char: 38 special char &

Fig ②:

remaining bytes that can be read: 2

n

104

0

105

remaining bytes that can be read: 0.

Fig ④:

number of bytes read: 2

bytes reading

✓ Sun Jan 03 2010
Sat Dec 02 03:21