

8/12/23

1) Print "Hello World"

→ public class PrintMain {

 public static void main (String args[]) {

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

 }

}

O/P → Hello World

2) Write a java program to add two numbers.

→ public class SumNum {

 public static void main (String args[]) {

 int a = 5;

 int b = 9;

 System.out.print (a+b);

 }

}

O/P → 14

3) Write a java program to multiply and divide two numbers.

→ public class Run1 {

 public static void main (String args[]) {

 int a = 6;

 int b = 2;

 System.out.print ("Product is = " + (a*b));

 if (b == 0) {

 System.out.print ("Not valid");

 }

 else {

 System.out.print (a/b);

 }

}

O/P → Product is = 12

3

23/12/2023

- 1) Write a program to overload the method print 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.

→ class Overload {

void print(int n) {

int sum = 0;

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

sum = sum + i;

}

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

}

void print(int m, int n) {

System.out.println("Prime numbers in 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.print(i);

}

}

}

```
public class OverloadDemo {  
    public static void main(String args[]) {  
        Overload o = new Overload();  
        o.print(5);  
        o.print(7, 13);  
    }  
}
```

Output

A Sum of 5 natural numbers is 15

B Prime numbers in the range are

PB
7
11
13

~~Ques~~

Q1) Write a Java program to create a class Grocery that has the variables c-name and c-phno. Create a method to accept 3 parameters to specify quantity of dd, quantity of tulsi 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-ph;

 double total;

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

 this.c-name = c-name;

 this.c-ph = c-ph;

}

 void calc (double q-dd, double q-tulsi, double
 q-sugar) {

$$\text{total} = q\text{-dd} * 100 + q\text{-tulsi} * 20 + q\text{-sugar} * 50;$$

}

 void display () {

 System.out.println ("Name" + " Ph.no number" +
 " Total");

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

 System.out.println ();

}

}

```

public class GroceryDemo {
    public static void main(String args[]) {
        Grocery g1 = new Grocery ("Rama", "8060302010");
        Grocery g2 = new Grocery ("Shama", "7689632510");
        Grocery g3 = new Grocery ("Bhama", "9632587412");

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

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

        g3.calc (1, 1, 0.5);
        g3.display ();
    }
}

```

| <u>Output</u> | Name | Phone Number | Total |
|---------------|-------|--------------|-------|
| | Rama | 8060302010 | 410.0 |
| | Shama | 7689632510 | 800.0 |
| | Bhama | 9632587412 | 205 |

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

→ import java.util.*;

class Quad {

int a, b, c;

double root1, root2, d;

Scanner s = new Scanner(System.in);

void input() {

System.out.print("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 discrimination() {

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

}

void calculateRoots() {

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("First root is: " + root1);

System.out.println("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 );
}

else {
    System.out.println( " Roots are imaginary. No real solution " );
    double real = -b / ( 2 * a );
    double imaginary = Math.sqrt ( -d ) / ( 2 * a );
    System.out.println( " The equation has two complex
roots: " + real + " + " + imaginary + "i and " +
real + " - " + imaginary + "i " );
}
}

```

```

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

```

```

        Quad q = new Quad ();
        q.input ();
        q.discriminant ();
        q.discriminant ();
        q.calculateRoots ();
    }
}

```

~~Q1 Q2 Q3 Q4 Q5 Q6~~

12/11/24

- 1) Create a class Book that contains four members : name, author, price and numPages. Include a constructor to set the values for the members. Include methods to set up and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a java program to create n book objects.

→ import java.util.*;

class Books {

String name;

String author;

int price;

int numPages;

Books (String name, String author, int price, int numPages){

this.name = name;

this.author = author;

this.price = price;

this.numPages = numPages;

}

public String toString() {

String name, author, price, numPages;

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

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

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

numPages = "number of Pages: " + this.numPages + "\n";

return bookName + AuthorName + bookPrice +

return Name + author + price + numPages;

}

}

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

Output

Enter the number of books: 2

Details of Book-1:

Enter the name of Book:-

abc

Enter the name of Author:-

xyz

Enter the price:-

500

Enter the number of pages:-

700

Details of Book-2:

Enter name of Book:-

def

Enter the name of Author:-

uvw

Enter the price:-

1000

Enter number of pages:-

600

Details of all books:

Book Name: abc

Author Name: xyz

Price : 500

Number of pages: 700

Book Name: def

Author Name: uvw

Price : 1000

Number of pages: 600

2) 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)

→ import java.util.*;

```
class Student {  
    String USN;  
    String name;  
    int marks[] = new int[6];  
  
    void acceptDetails() {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter USN: ");  
        USN = sc.nextLine();  
        System.out.print("Enter name: ");  
        name = sc.nextLine();  
        System.out.println("Enter marks of for 6 subjects: ");  
        for (int i=0; i<6; i++) {  
            System.out.print("Subject " + (i+1) + ": ");  
            marks[i] = sc.nextInt();  
        }  
  
        double calculatePercentage() {  
            int totalMarks = 0;  
            for (int mark : marks) {  
                totalMarks += mark;  
            }  
            return (double) totalMarks / 6;  
        }  
}
```

```
Void display Details() {  
    System.out.println("Student details:");  
    System.out.println("USN " + USN);  
    System.out.println("Name " + name);  
    System.out.println("Percentage: " + calculate Percentage());  
}
```

```
public class StudentDot {
```

```
    public static void main (String args[]) {  
        Scanner sc = new Scanner (System.in);  
        System.out.print ("Enter the number of students: ");  
        int numStudents = Scanner sc.nextInt();  
        Student students[] = new Student [numStudents];  
        for (int i=0; i<numStudents; i++) {  
            Student[i] = new Student();  
            System.out.println ("Enter details for student " +  
                (i+1) + ":");  
            student[i].accept Details();  
        }  
        System.out.println ("Details of all students are: ");  
        for (Student student : students) {  
            student.display Details();  
            System.out.println ();  
        }  
    }  
}
```

Output

Enter the number of students: 21

Enter details of student 1:

Enter USN: 1BM22CS001

Enter Name: Aaditya

Enter marks for 6 subjects:

Subject 1: 50

Subject 2: 60

Subject 3: 70

Subject 4: 80

Subject 5: 90

Subject 6: 95

~~70 50 60 80 90 95~~

Student details:

USN: 1BM22CS001

Name: Aaditya

Marks:

Subject 1: 50.00

Subject 2: 60.00

Subject 3: 70.00

Subject 4: 80.00

Subject 5: 90.00

Subject 6: 95.00

Percentage: 74.166667

Program to calculate percentage

Q1/12

1) Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain the method printArea() that prints the area of the given shape.

→ abstract class Shape {

 int side1;

 int side2;

 Shape (int side1, int side2) {

 this.side1 = side1;

 this.side2 = side2;

}

 public abstract void printArea();

}

class Rectangle extends Shape {

 Rectangle (int length, int width) {

 super(length, width);

}

 public void printArea() {

 int area = side1 * side2;

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

}

}

class Triangle extends Shape {

 Triangle (int base, int height) {

 super(base, height);

}

```
public void printArea() {
    int area = side1 * side2;
    System.out.println("Area of triangle: " + area);
}
```

```
class Circle extends Shape {
    Circle(int radius) {
        Super(radius, 0);
    }
}
```

```
public void printArea() {
    int area = Math.PI * side1 * side2;
}
```

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

```

```
        Rectangle r1 = new Rectangle (5, 8);

```

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

```

```
        Circle c1 = new Circle (3);

```

```
        r1.printArea();

```

```
        t1.printArea();

```

```
        c1.printArea();

```

Output

Area of rectangle: 40

Area of triangle: 12.0

Area of circle: 28.274333882308138

2) Develop a java program to create a class 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 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 if and if 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 derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following task:

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

Check for minimum balance, impose penalty if necessary and update the balance.

> import java.util.*;

class Account {

String customerName;

long accountNumber;

String accountType;

double balance;

public Account(String customerName, long accountNumber,
String accountType, double balance) {

this.accountNumber = accountNumber;

this.customerName = customerName;

this.accountType = accountType;

this.balance = balance;

}

public void displayBalance() {

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

System.out.print("Customer Name: " + customerName);

System.out.print("Account type: " + accountType);

System.out.print("Balance: \$" + balance);

}

public void deposit(double amount) {

~~if (amount <= balance) {~~

balance = balance + amount;

System.out.println("Deposit of \$" + amount + " successful");

~~else { displayBalance(); }~~

~~System.out.println("Insufficient funds");~~

~~public void withdrawal(double amount) {~~

~~if (amount <= balance) {~~

~~balance = amount + balance;~~

~~System.out.println("Withdrawal of \$" + amount + "~~

~~"successful"); }~~

~~else {~~

~~System.out.println("Insufficient funds. Withdrawal failed");~~

~~? slightly Det. 1.1.2 ?~~

class CurrentAccount extends Amount {

 double minBal = 1000;

 double serviceCharge = 50;

 CurrentAccount (String customerName, long accountNumber,
 double balance) {

 super (customerName, accountNumber, "Current",
 balance);

}

 public void withdrawal (double amount) {

 if (amount <= balance - minBal) {

 balance = balance - amount;

 System.out.println ("Withdrawal of " + amount + " success");

 } else {

 System.out.println ("Insufficient funds. Withdrawal failed.");

 service charge of " + serviceCharge + " imposed.");

 balance = balance - serviceCharge;

 }

 displayBalance ();

 }

}

class SavingsAccount extends Amount {

 double interestRate = 0.05;

 public SavingsAccount (String customerName, long accountNumber,
 double balance) {

 super (customerName, accountNumber, "Savings", balance);

}

public void computeInterest () {

$$\text{double interest} = \text{balance} * \text{interest};$$

$$\text{balance} = \text{balance} + \text{interest};$$

System.out.println ("Interest of " + interest + " credited");

displayBalance ();

}

public class TestC

public static void main (String args[]) {

Scanner sc = new Scanner (System.in);

Customer s1 = new Customer ("Aman", 123456789, 5000);

s1.displayBalance();

s1.deposit(1000);

s1.computeInterest();

s1.withdrawal(2000);

Customer c1 = new Customer ("Aditya", 987654321, 1500);

c1.displayBalance();

c1.deposit(500);

c1.withdrawal(2000);

}

3

✓

Output

Account Number : 123456789

Customer Name : Aman

Account Type : Savings

Balance : 5000

Deposit of 1000 successful.

Account Number : 123456789

Customer Name : Aman

Account Type : Savings

Balance : 6000

Interest of 300 credited

Account Number : 123456789

Customer Name : Aman

Account type : Savings

Balance : 6300

Withdrawal of 2000 successful.

Account Number : 123456789

Customer Name : Aman

Account Type : Savings

Balance : 4300

Account Number : 987654321

Customer Name : Aditya

Balance : 1500

~~Withdrawal of~~

~~Deposit of 500 successful~~

Account Number : 987654321

~~Customer Name : Aditya~~

Balance : 2000

~~Inufficient funds. Withdrawal failed. Service charge of 50 imposed.~~

Account Number: 987654321

Customer Name: Aditya

Account Type: Current

Balance: 1950

~~CSL
19.01.21~~

Q6

Create a package CIE which has two classes - Student and Internals. The class Student has members like USN, name, sem. The class internal, marks scored in five courses of the current semester of students. Create another class SEE which has the class Internal which is a derived class of student. Create This class has an array that stores the SEE marks scored in five courses of the current semester of the students.

→

```
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 sem = sc.nextInt();  
    }  
}
```

```
package CTE;
import java.util.Scanner;
protected int marks[] = new int[5];
public void inputCTEmarks() {
    Scanner sc = new Scanner(System.in);
    for(int i=0 ; i<5 ; i++){
        System.out.println("Enter marks for course " + (i+1));
        marks[i] = sc.nextInt();
    }
}
package SEE;
import CTE.internals;
import java.util.Scanner;
public class External extends Internals {
    protected int marks[];
    protected int finalMarks[];
}
for class External() {
    marks = new int[5];
    finalMarks = new int[5];
}
protected void inputSEEmarks(){
    Scanner sc = new Scanner(System.in);
    for(int i=0 ; i<5 ; i++){
        System.out.println("Subject " + (i+1) + " marks:");
        marks[i] = sc.nextInt(); } }
```

```
public void calcFinalMarks () {  
    for (i=0; i<5)  
        finalMarks[i] = marks[i] + subjectMarks[i]  
    }  
}
```

```
package SFF;  
public class External extends IInternal {  
    public External();  
    public class Main {  
        public static void main (String ar[]) {  
            for (i=0; i<numofStudents; i++) {  
                finalMarks[i] = new External();  
                finalMarks[i] = new External();  
                finalMarks[i] = input (IEMarks);  
            }  
        }  
    }
```

```
System.out.println ("display data");
```

```
for (int i=0; i<no.of students; i++) {  
    finalMarks[i].calcFinalMarks();  
}  
}
```

class External implements IInternal {
 int marks[] = {10, 20, 30, 40, 50};
 int subjectMarks[] = {10, 20, 30, 40, 50};
 public void calcFinalMarks () {
 for (i=0; i<5) {
 finalMarks[i] = marks[i] + subjectMarks[i];
 }
 }
}

(External is a child class of IInternal)

class Main {
 public static void main (String ar[]) {
 External obj = new External();
 obj.calcFinalMarks();
 }
}

Q) 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 age ≤ 0 . In son Son class, implement a constructor that uses both father and son's age and throws an exception if son's age is \geq father's age.

-> import java.util.Scanner;

class WrongAge extends Exception {

 public WrongAge() {

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

}

 public WrongAge(String message) {

 super(message);

}

}

class InputScanner {

 public static int readInt() {

 try { Scanner scanner = new Scanner(System.in); }

 catch { return scanner.nextInt(); }

 }

}

class Father extends InputScanner {

 protected int fatherAge;

 public Father() throws WrongAge {

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

```
fatherAge = readInt();
```

```
if (fatherAge < 0) {
```

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

```
}
```

```
public void display() {
```

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

```
}
```

```
class Son extends Father {
```

```
protected int sonAge;
```

```
public Son() throws WrongAge {
```

```
super();
```

```
System.out.print ("Enter son's age: ");
```

```
sonAge = readInt();
```

```
if (sonAge >= super.fatherAge) {
```

```
    throw new WrongAge ("Son's age
```

```
    cannot be greater than father's");
```

```
}
```

```
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 Exception {
    public static void main(String a[]) {
        try {
            Son son = new Son();
            son.display();
        } catch (WrongAge e) {
            System.out.println("error: " + e.getMessage());
        }
    }
}

```

Q) Write a program which creates two threads,
one thread displaying "BMS College of Engineering"
once every 10 seconds and another displaying
"CSE" once every two seconds.

```

→ class DisplayMessage extends Thread {
    private String message;
    private int interval;

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

    public void run() {
        while(true) {
            System.out.println(message);
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

```

public class Threading {

 public static void main (String a[]) {

 DisplayMessage t1 = new DisplayMessage ("

 BMS College of Engineering ", 10);

 DisplayMessage t2 = new DisplayMessage ("CSI
 2");

 thread1.start();

 thread2.start();

 }

Output : BMS College of Engineering

BMS College of Engineering

CSE

BMS College of Engineering

CSE

CSE

CSE

BMS College of Engineering

~~CS1602~~

23/02/24

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

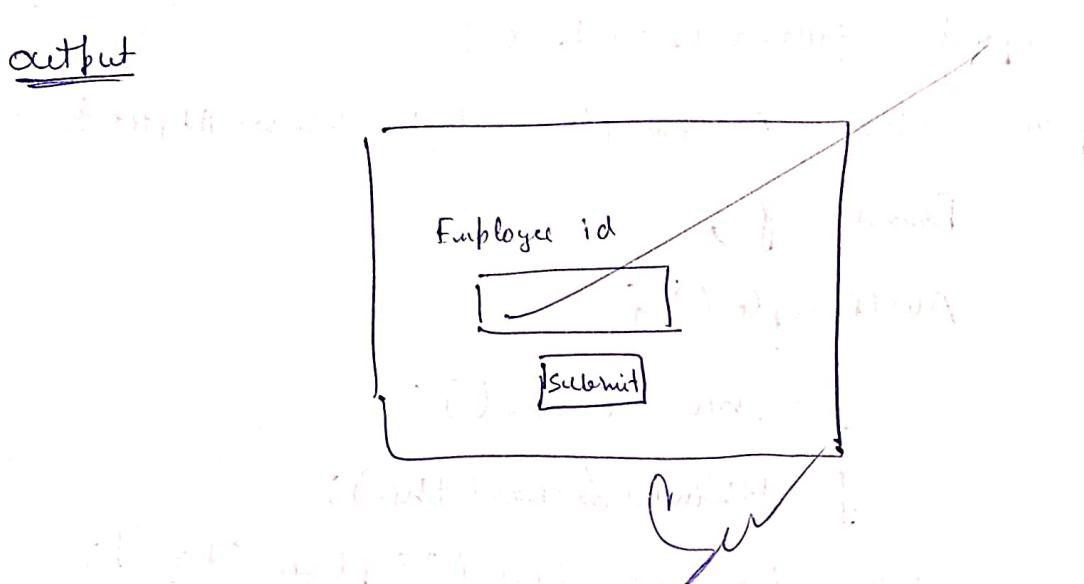
```
→ import java.awt.*;  
import java.awt.event.*;  
public class AWTExample extends WindowAdapter {  
    Frame f;  
    AWTExample () {  
        f = new Frame();  
        f.addWindowListener(this);  
        Label l = new Label ("Employee id:");  
        Button b = new Button ("Submit");  
        Textfield t = new Textfield();  
        Textfield t = new Textfield();  
        l.setBounds (20, 80, 80, 30);  
        t.setBounds (20, 100, 80, 30);  
        b.setBounds (100, 100, 80, 30);  
        f.add(b);  
        f.add(l);  
        f.add(t);  
        f.setSize (400, 300);  
        f.setTitle ("Employee info");  
        f.setLayout (null);  
        f.setVisible (true);  
    }  
    public void windowClosing (WindowEvent e) {  
        System.exit(0);  
    }
```

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

```
    AWTExample awtObj = new AWTExample();
```

```
}
```

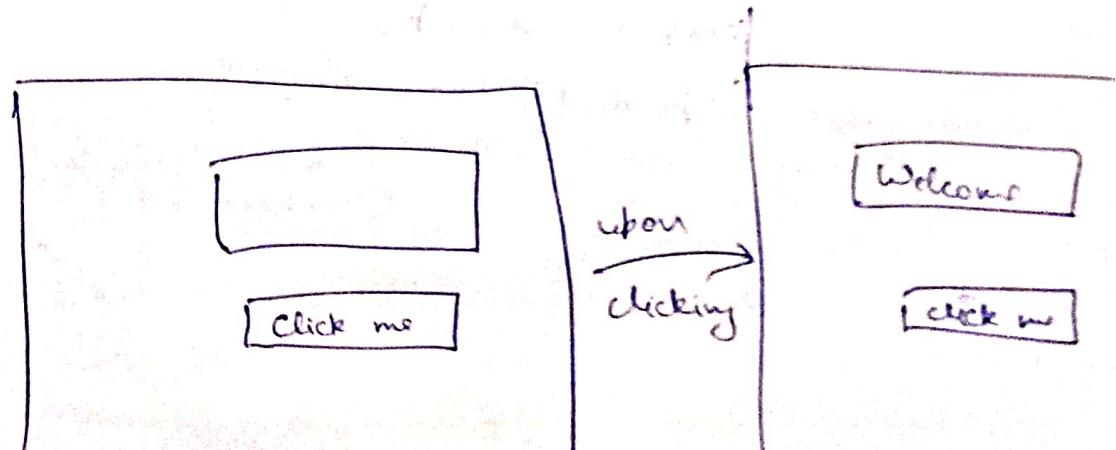
~~Output~~



2) Create a button and add a listener for Mouse click

```
→ import java.awt.*;  
import java.awt.event.*;  
public class EventHandling extends WindowAdapter  
implements ActionListener {  
  
    Frame f;  
    f.addWindowListener(this);  
    f = new TextField();  
    f.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.setLayout(null);  
    f.setVisible(true);  
}  
  
public void windowClosing(WindowEvent e) {  
    System.exit(0);  
}  
  
public static void main (String args []){  
    new EventHandling();  
}
```

Output



Example - 1

```

import java.io;
public class ByteArrayInput {
    public static void main( String a[] ) throws
        IOException {
        byte [ ] buf = { 35, 36, 37, 38 };
        ByteArrayInputStream byt = new ByteArrayInputStream( buf );
        int k = 0;
        while ( ( k = byt.read() ) != -1 ) {
            char ch = (char) k ;
            System.out.println( " ASCII value of character is : "
                + k + " ; special character is : " + ch );
        }
    }
}

```

Example - 2

```

import java.io.*;
public class ByteArrayInput {
    public static void main( String a[] ) throws IOException {
        byte buf [ ] = { 35, 36, 37, 38 };
        ByteArrayInputStream byt = new ByteArrayInputStream( buf );
        int k = 0;
        while ( ( k = byt.read() ) != -1 ) {
            char ch = (char) k;
            System.out.println( " ASCII value of character is : "
                + k + " ; special character is : " + ch );
        }
    }
}

```

Ans

Example 3

```
public class FileFn {  
    public static void main(String a[]) 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.println ((char) content + " ");  
        System.out.println (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 FileFn2 {  
    public static void main(String a[]) throws IOException {  
        FileInputStream fin = new FileInputStream ("Example.txt");  
        byte [] bytes = new byte [20];  
        int i;  
        char c;  
  
        i = fin.read (bytes);  
        System.out.println ("Number of bytes read: " + i);  
        System.out.println ("Bytes read: ");  
  
        for (byte b: bytes) {  
            c = (char) b;  
            System.out.println (c);  
        }  
    }  
}
```

Example 5

```
import java.io.*;  
public class ByteArray_ew {  
    public static void main(String a[]) throws Exception {  
        FileOutputStream fout1 = new FileOutputStream("Example1.txt");  
        FileOutputStream fout2 = new FileOutputStream("Example2.txt");  
        ByteArrayOutputStream bout = new ByteArrayOutputStream();  
        bout.write(65);  
        bout.writeTo(fout1);  
        bout.writeTo(fout2);  
        bout.flush();  
        bout.close();  
        System.out.println("Success...");  
    }  
}
```

O/P

- 1) Ascii of char : 35 ; special char : #
- 2) Ascii of char 36 ; special char \$
- 3) Ascii of char 37 ; special char %
- 4) Ascii of char 38 ; special char &

- 3) Remaining bytes that can be read : 2

h
104
i
105

remaining bytes that can be read : 0

47 Number of bytes read : 2

Bytes read : hi

~~QFDL
JPG
23.02.2011~~