MAHARAJA SURAJMAL INSTITUTE

C-4, Janakpuri, New-Delhi, 110058

Department of Computer Applications

JAVA PRACTICAL FILE

Course Code: **BCA 272**

Course Name: **Practical-VII JAVA Lab**

Submitted By:

Name: AKANKSHA DANGRI
Enrolment Number: 08914902022
Semester and Section: IV - B

Submitted To:

Mr. Manoj Kumar
Asst. Professor

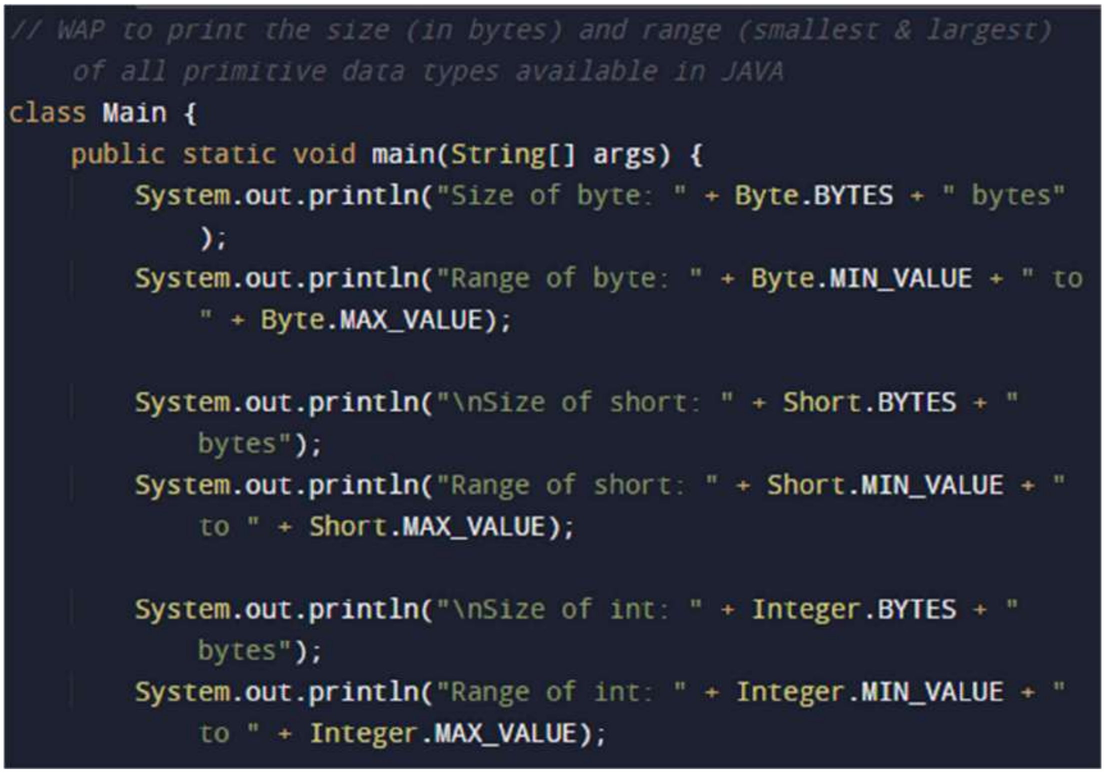
(Affiliated to GGSIP University)
2022-25

INDEX

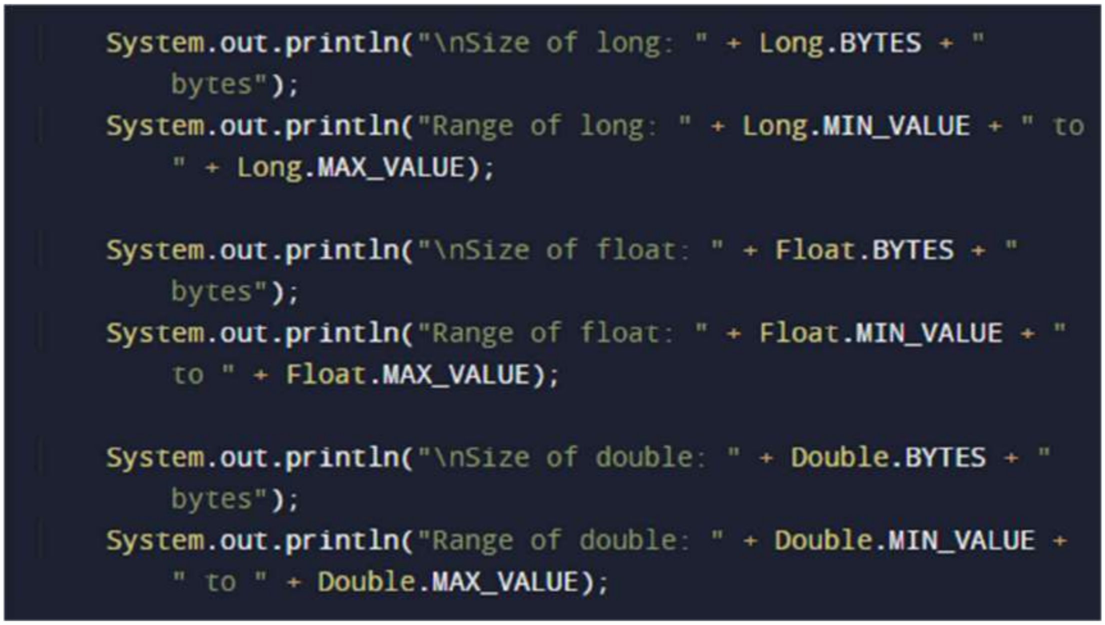
Application Based Practicals

32	Create a class employee which have name, age and address of employee, include methods getdata() and showdata(), getdata() takes the input from the user, showdata() display the data in following format: Name: Age: Address:	
33	WAP to perform basic Calculator operations. Make a menu driven program to select operation to perform (+ - * /). Take 2 integers and perform operation as chosen by user.	
34	WAP to make use of BufferedStream to read lines from the keyboard until 'STOP' is typed.	
35	WAP declaring a Java class called SavingsAccount with members ``accountNumber`` and ``Balance``. Provide member functions as ``depositAmount ()`` and ``withdrawAmount ()``. If user tries to withdraw an amount greater than their balance then throw a user-defined exception.	
36	WAP creating 2 threads using Runnable interface. Print your name in ``run ()`` method of first class and "Hello Java" in ``run ()`` method of second thread.	
37	Write program that uses swings to display combination of RGB using 3 scrollbars.	
38	Write a swing application that uses at least 5 swing controls.	
39	Write a program to implement border layout using Swing.	
40	Write a java program to insert and update details data in the database.	
41	Write a java program to retrieve data from database and display it on GUI.	

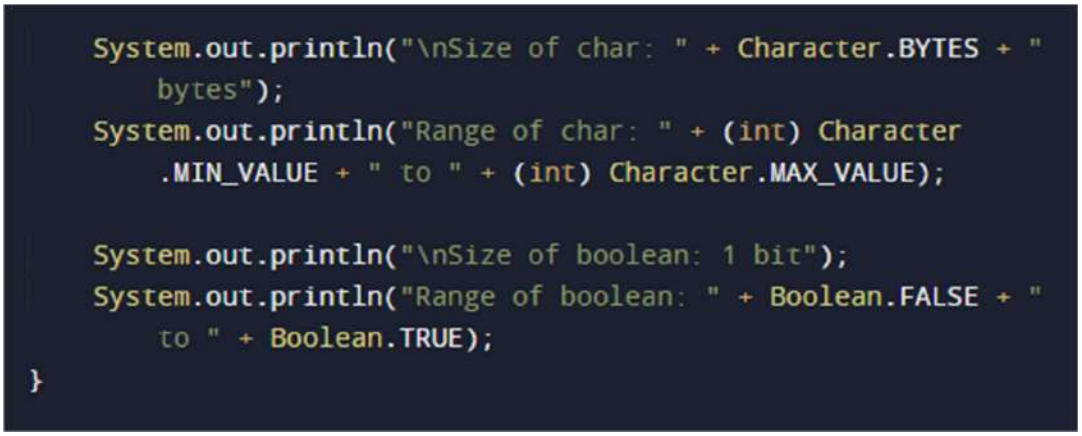
Ques1: WAP to print the size (in bytes) and range (smallest & largest) of all primitive data types available in JAVA.



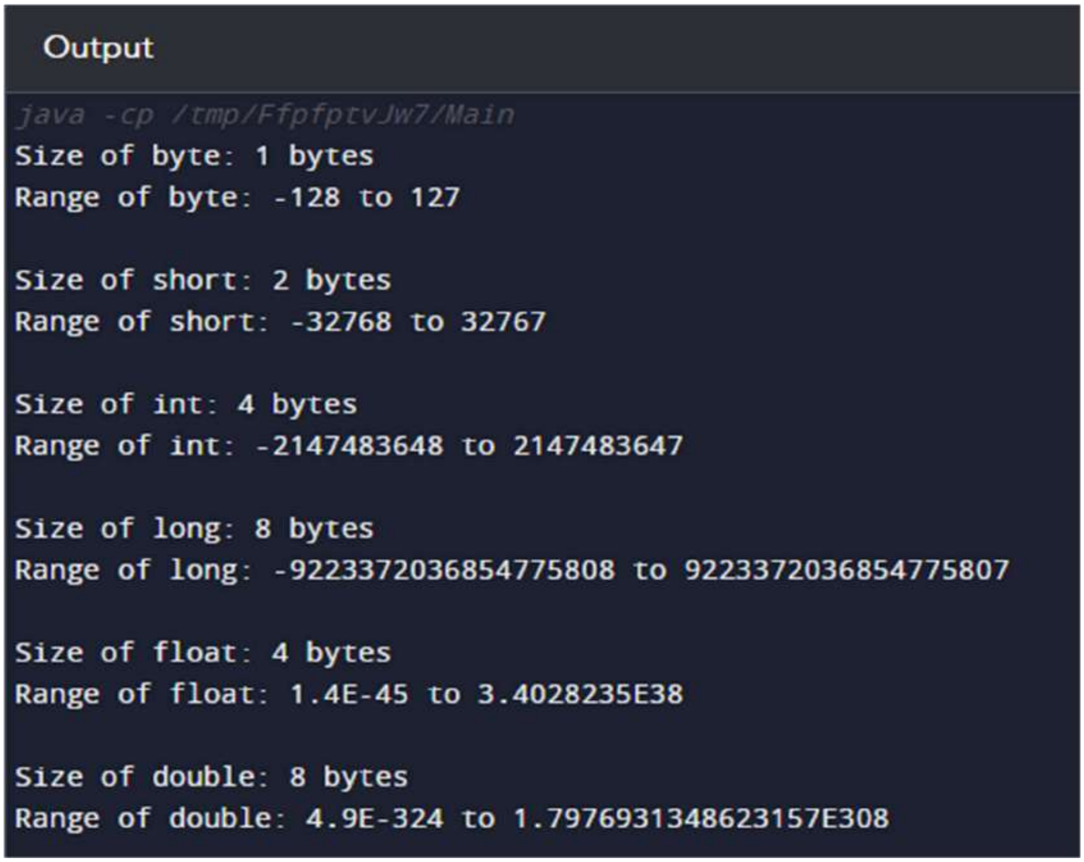
```
// WAP to print the size (in bytes) and range (smallest & largest)
    of all primitive data types available in JAVA
class Main {
    public static void main(String[] args) {
        System.out.println("Size of byte: " + Byte.BYTES + " bytes"
            );
        System.out.println("Range of byte: " + Byte.MIN_VALUE + " to
            " + Byte.MAX_VALUE);

        System.out.println("\nSize of short: " + Short.BYTES + "
            bytes");
        System.out.println("Range of short: " + Short.MIN_VALUE + "
            to " + Short.MAX_VALUE);

        System.out.println("\nSize of int: " + Integer.BYTES + "
            bytes");
        System.out.println("Range of int: " + Integer.MIN_VALUE + "
            to " + Integer.MAX_VALUE);
```



```
        System.out.println("\nSize of long: " + Long.BYTES + "
            bytes");
        System.out.println("Range of long: " + Long.MIN_VALUE + " to
            " + Long.MAX_VALUE);

        System.out.println("\nSize of float: " + Float.BYTES + "
            bytes");
        System.out.println("Range of float: " + Float.MIN_VALUE + "
            to " + Float.MAX_VALUE);

        System.out.println("\nSize of double: " + Double.BYTES + "
            bytes");
        System.out.println("Range of double: " + Double.MIN_VALUE +
            " to " + Double.MAX_VALUE);
```

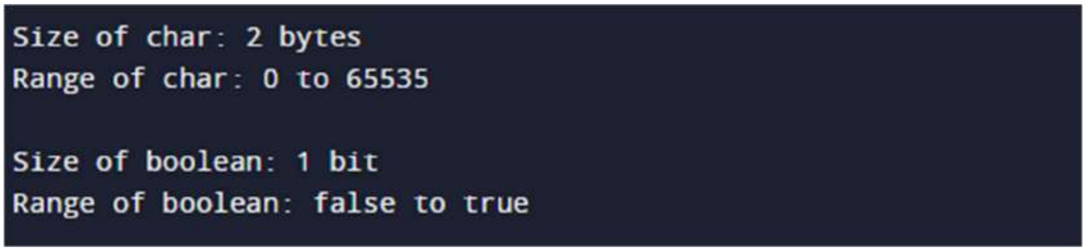


```
    System.out.println("\nSize of char: " + Character.BYTES + "
        bytes");
    System.out.println("Range of char: " + (int) Character
        .MIN_VALUE + " to " + (int) Character.MAX_VALUE);

    System.out.println("\nSize of boolean: 1 bit");
    System.out.println("Range of boolean: " + Boolean.FALSE + "
        to " + Boolean.TRUE);
}
```



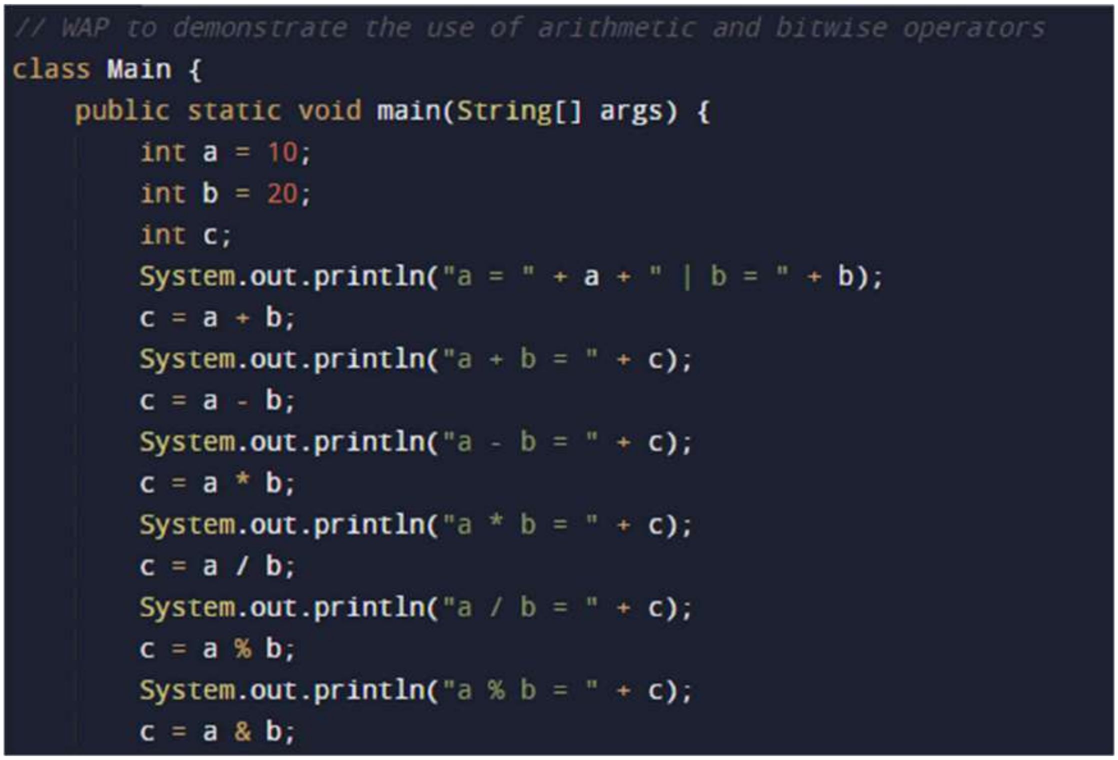
Output

```
java -cp /tmp/FfpfptvJw7/Main
Size of byte: 1 bytes
Range of byte: -128 to 127

Size of short: 2 bytes
Range of short: -32768 to 32767

Size of int: 4 bytes
Range of int: -2147483648 to 2147483647

Size of long: 8 bytes
Range of long: -9223372036854775808 to 9223372036854775807

Size of float: 4 bytes
Range of float: 1.4E-45 to 3.4028235E38

Size of double: 8 bytes
Range of double: 4.9E-324 to 1.7976931348623157E308
```

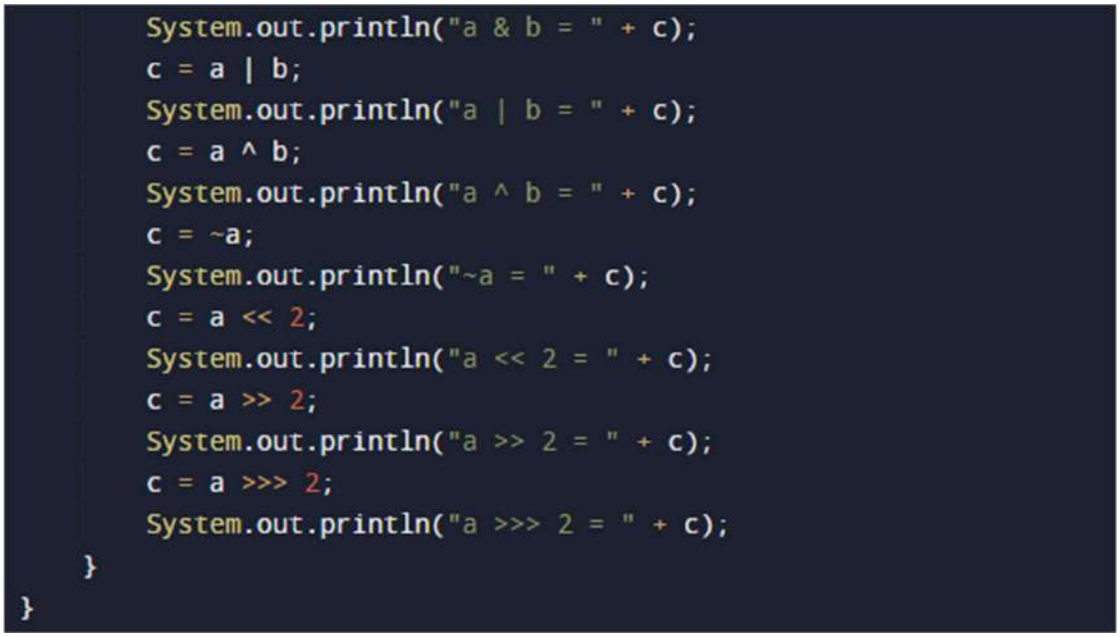


```
Size of char: 2 bytes
Range of char: 0 to 65535

Size of boolean: 1 bit
Range of boolean: false to true
```

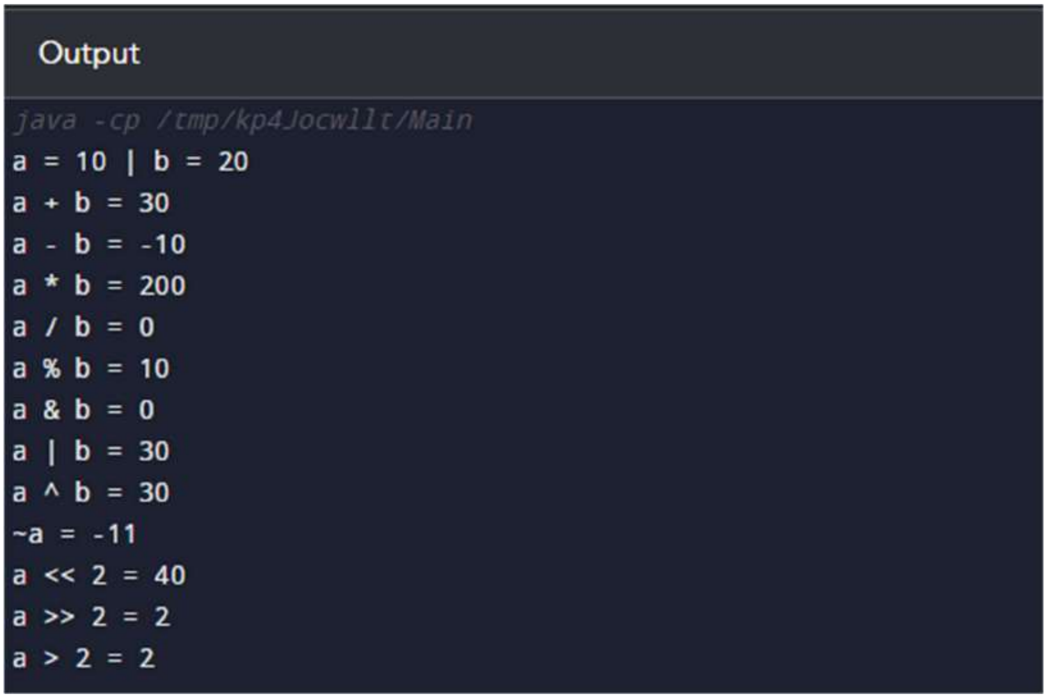
Ques2: WAP to demonstrate the use of arithmetic and bitwise operators.



```
// WAP to demonstrate the use of arithmetic and bitwise operators
class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 20;
        int c;
        System.out.println("a = " + a + " | b = " + b);
        c = a + b;
        System.out.println("a + b = " + c);
        c = a - b;
        System.out.println("a - b = " + c);
        c = a * b;
        System.out.println("a * b = " + c);
        c = a / b;
        System.out.println("a / b = " + c);
        c = a % b;
        System.out.println("a % b = " + c);
        c = a & b;
```

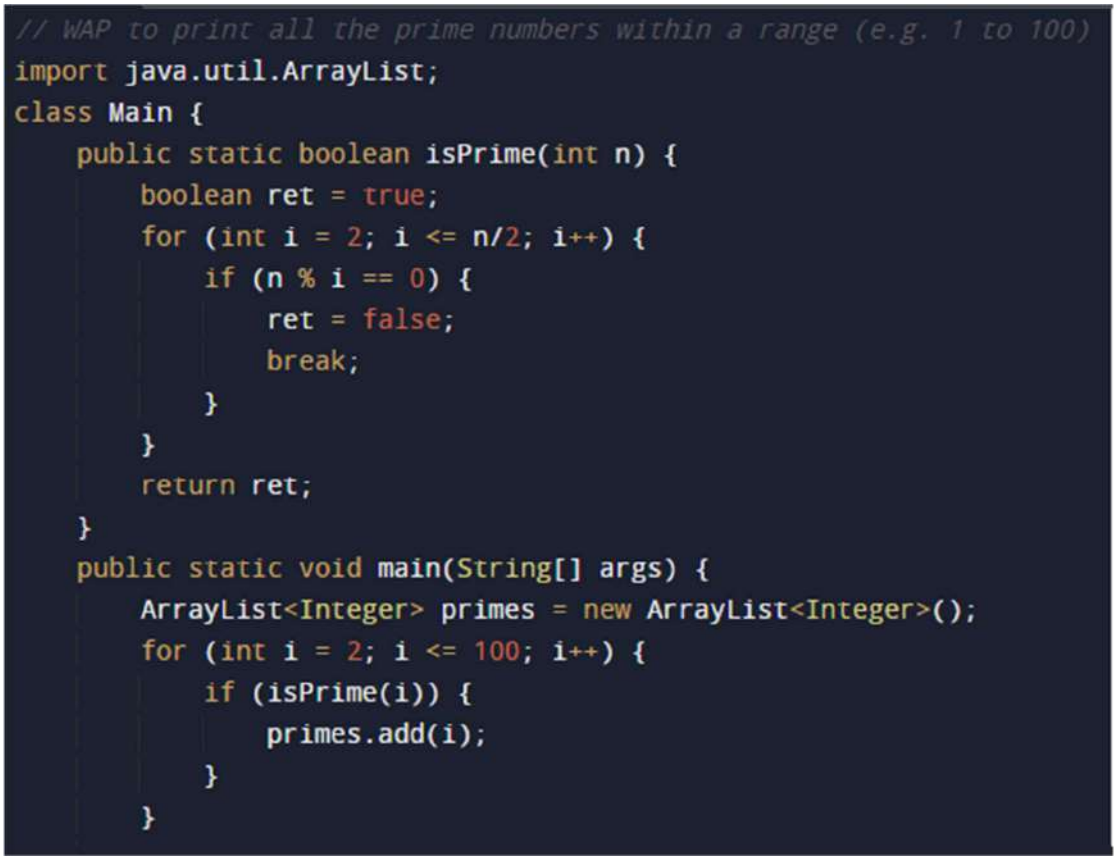


```
        System.out.println("a & b = " + c);
        c = a | b;
        System.out.println("a | b = " + c);
        c = a ^ b;
        System.out.println("a ^ b = " + c);
        c = ~a;
        System.out.println("~a = " + c);
        c = a << 2;
        System.out.println("a << 2 = " + c);
        c = a >> 2;
        System.out.println("a >> 2 = " + c);
        c = a >>> 2;
        System.out.println("a >>> 2 = " + c);
    }
}
```

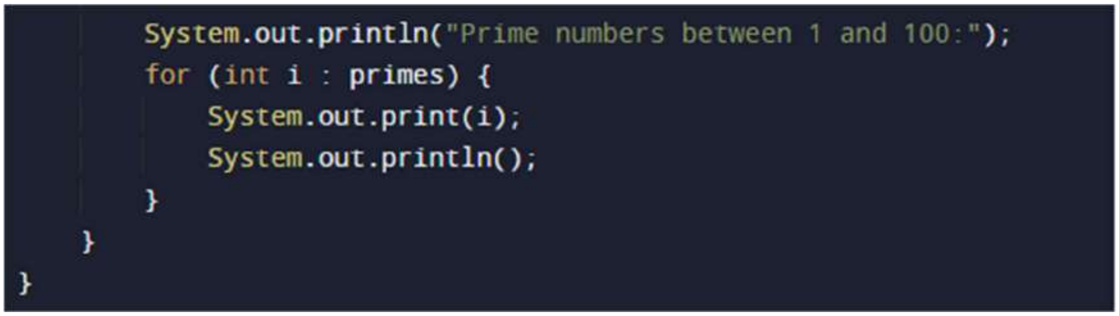


```
java -cp /tmp/kp4Jocwllt/Main
a = 10 | b = 20
a + b = 30
a - b = -10
a * b = 200
a / b = 0
a % b = 10
a & b = 0
a | b = 30
a ^ b = 30
~a = -11
a << 2 = 40
a >> 2 = 2
a > 2 = 2
```

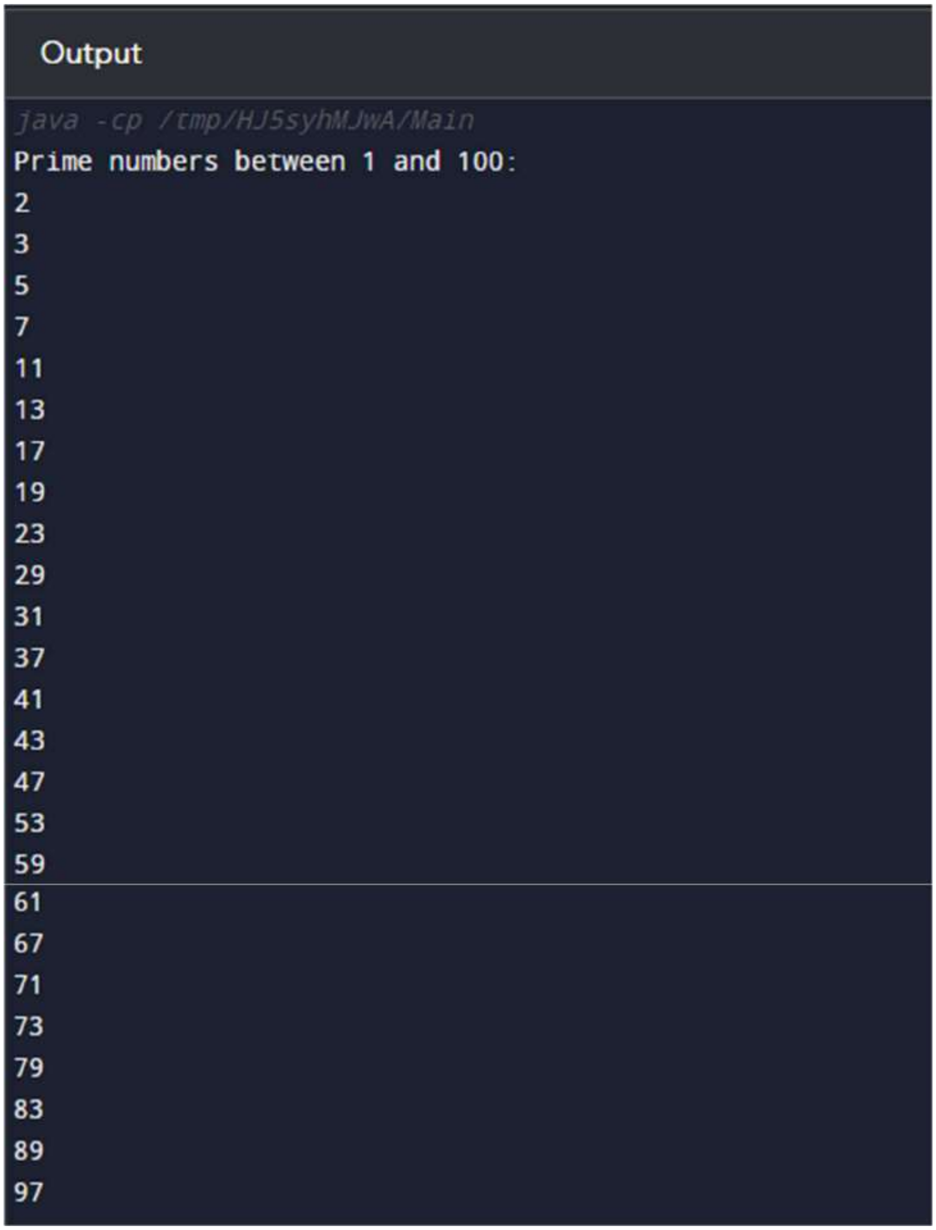
Ques3: WAP to print all the prime numbers within a range (e.g. 1 to 100).



```java
// WAP to print all the prime numbers within a range (e.g. 1 to 100)
import java.util.ArrayList;
class Main {
    public static boolean isPrime(int n) {
        boolean ret = true;
        for (int i = 2; i <= n/2; i++) {
            if (n % i == 0) {
                ret = false;
                break;
            }
        }
        return ret;
    }
    public static void main(String[] args) {
        ArrayList<Integer> primes = new ArrayList<Integer>();
        for (int i = 2; i <= 100; i++) {
            if (isPrime(i)) {
                primes.add(i);
            }
        }
```

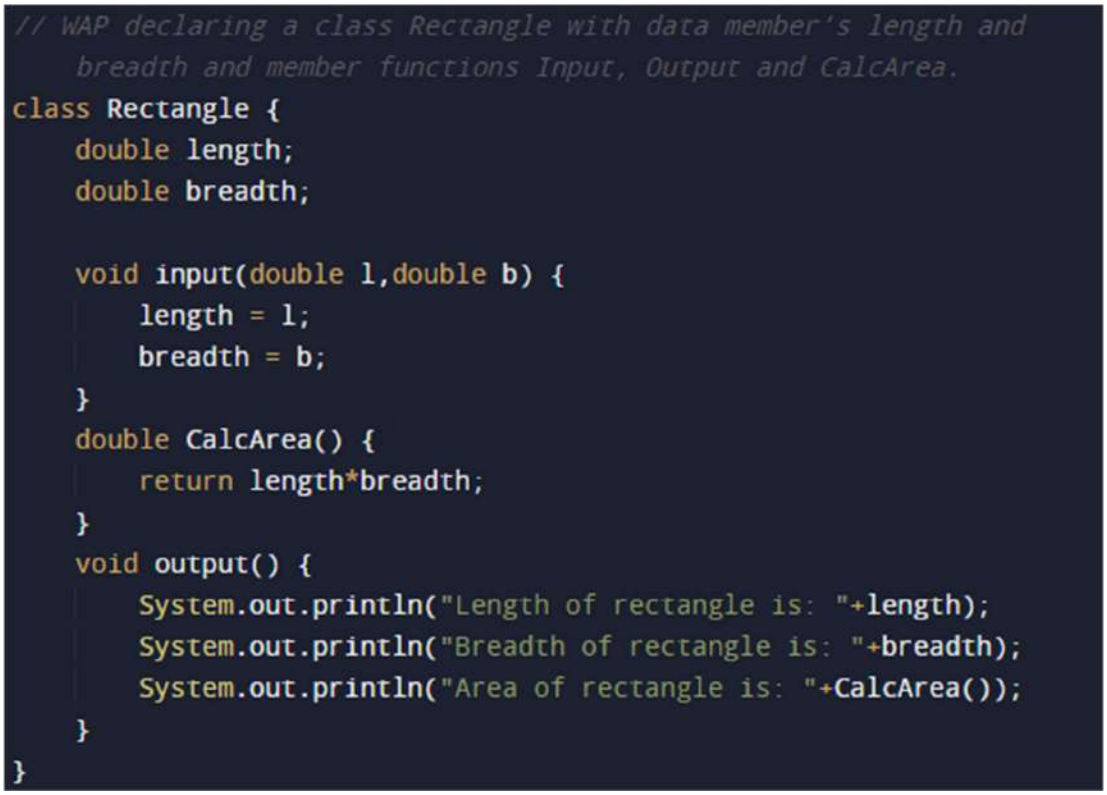


```java
        System.out.println("Prime numbers between 1 and 100:");
        for (int i : primes) {
            System.out.print(i);
            System.out.println();
        }
    }
}
```

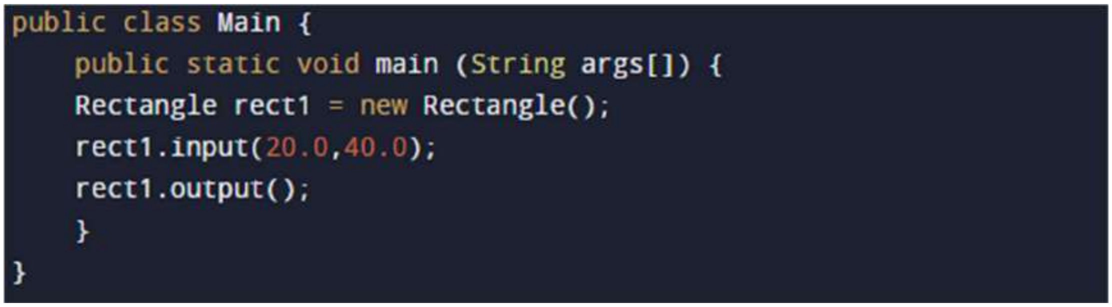


```
java -cp /tmp/HJ5syhMJwA/Main
Prime numbers between 1 and 100:
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
```

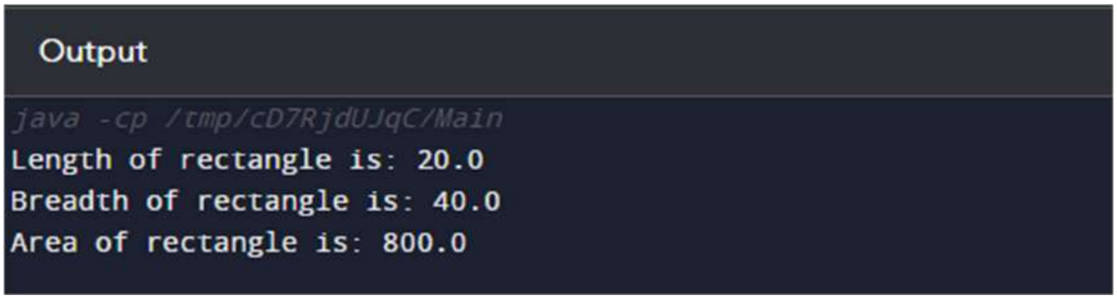
Ques4: WAP declaring a class Rectangle with data member’s length and breadth and member functions Input, Output and CalcArea.



```java
// WAP declaring a class Rectangle with data member’s length and
    breadth and member functions Input, Output and CalcArea.
class Rectangle {
    double length;
    double breadth;

    void input(double l,double b) {
        length = l;
        breadth = b;
    }
    double CalcArea() {
        return length*breadth;
    }
    void output() {
        System.out.println("Length of rectangle is: "+length);
        System.out.println("Breadth of rectangle is: "+breadth);
        System.out.println("Area of rectangle is: "+CalcArea());
    }
}
```



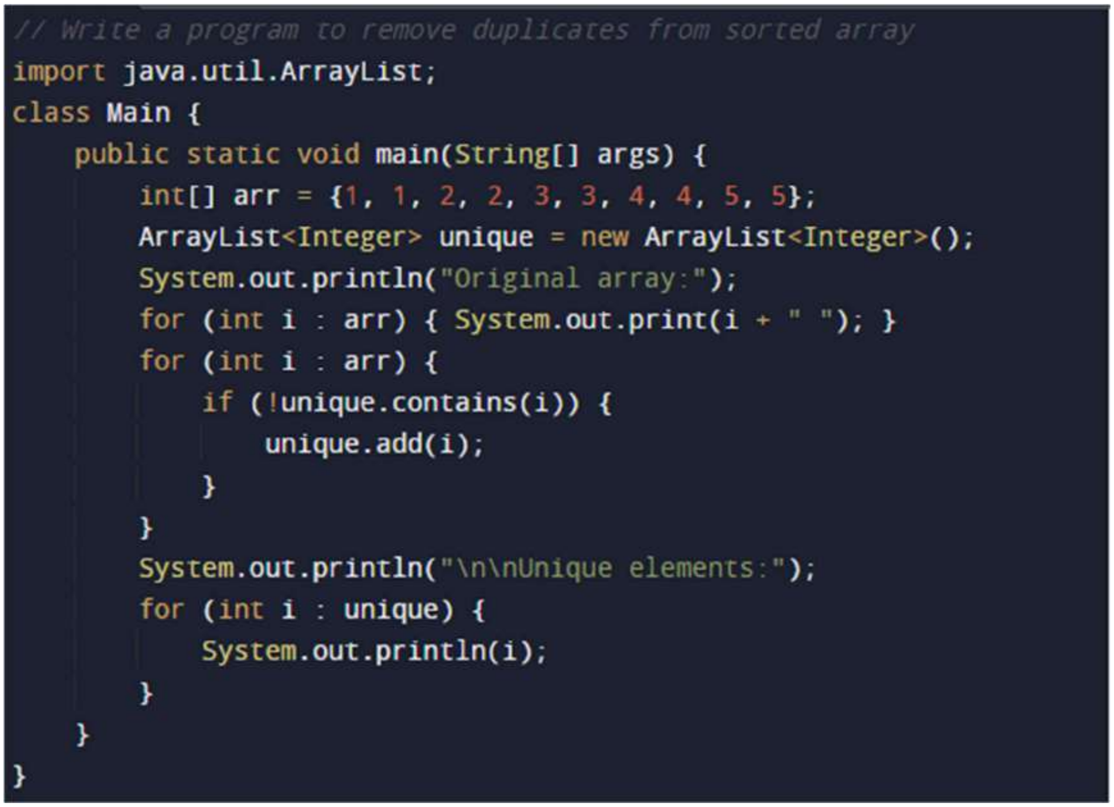
```java
public class Main {
    public static void main (String args[]) {
    Rectangle rect1 = new Rectangle();
    rect1.input(20.0,40.0);
    rect1.output();
    }
}
```



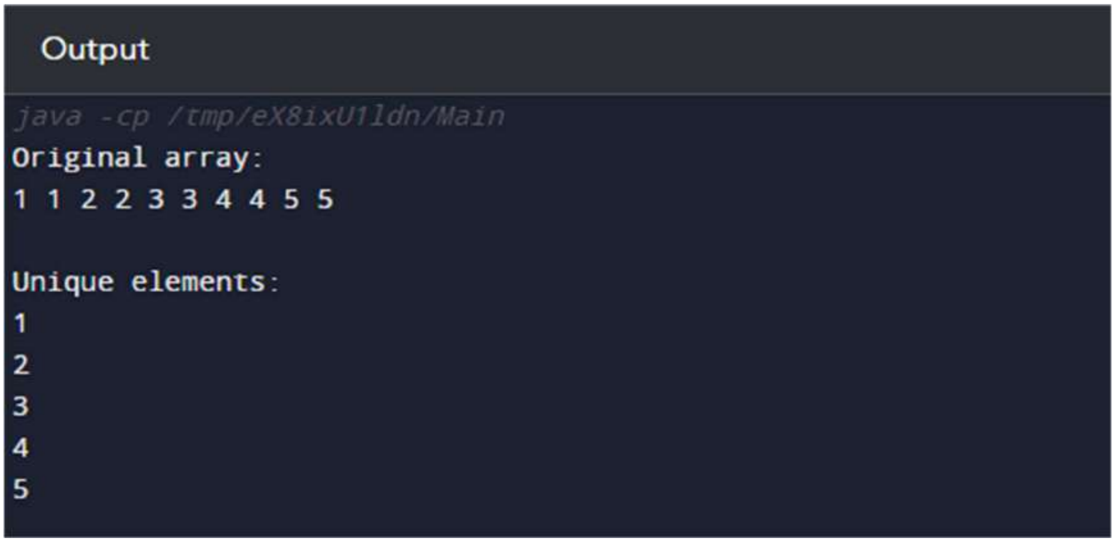
Output

```
java -cp /tmp/cD7RjdUJqC/Main
Length of rectangle is: 20.0
Breadth of rectangle is: 40.0
Area of rectangle is: 800.0
```

Ques5: Write a program to remove duplicates from sorted array.

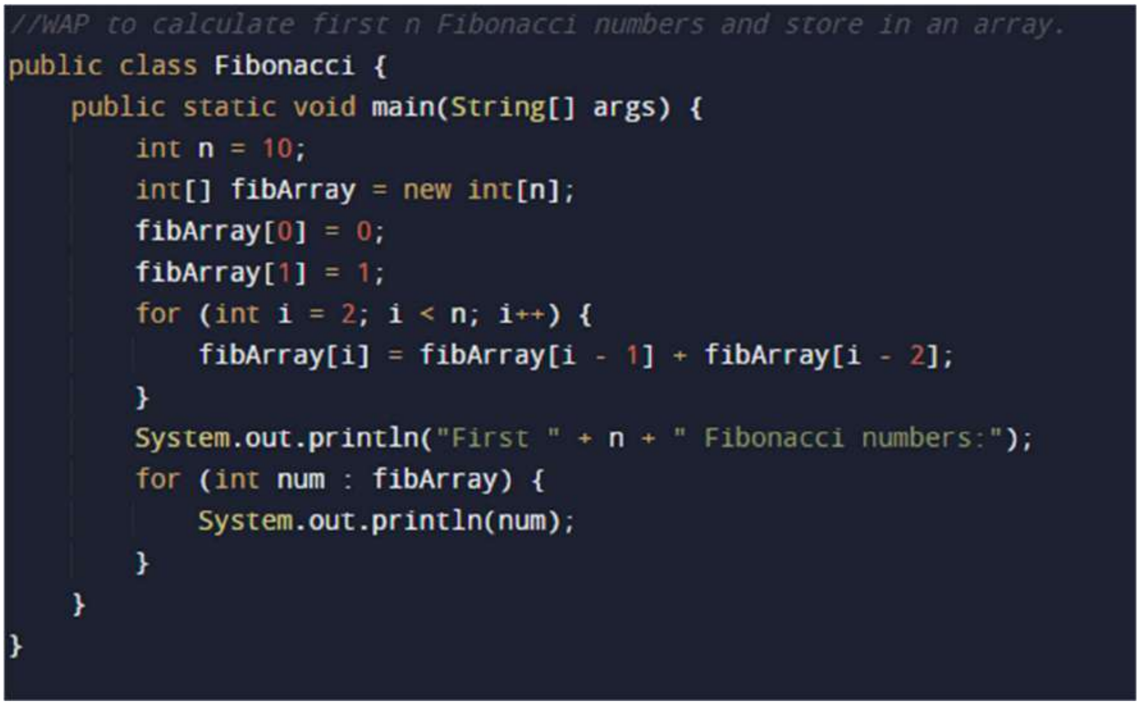


```java
// Write a program to remove duplicates from sorted array
import java.util.ArrayList;
class Main {
    public static void main(String[] args) {
        int[] arr = {1, 1, 2, 2, 3, 3, 4, 4, 5, 5};
        ArrayList<Integer> unique = new ArrayList<Integer>();
        System.out.println("Original array:");
        for (int i : arr) { System.out.print(i + " "); }
        for (int i : arr) {
            if (!unique.contains(i)) {
                unique.add(i);
            }
        }
        System.out.println("\n\nUnique elements:");
        for (int i : unique) {
            System.out.println(i);
        }
    }
}
```

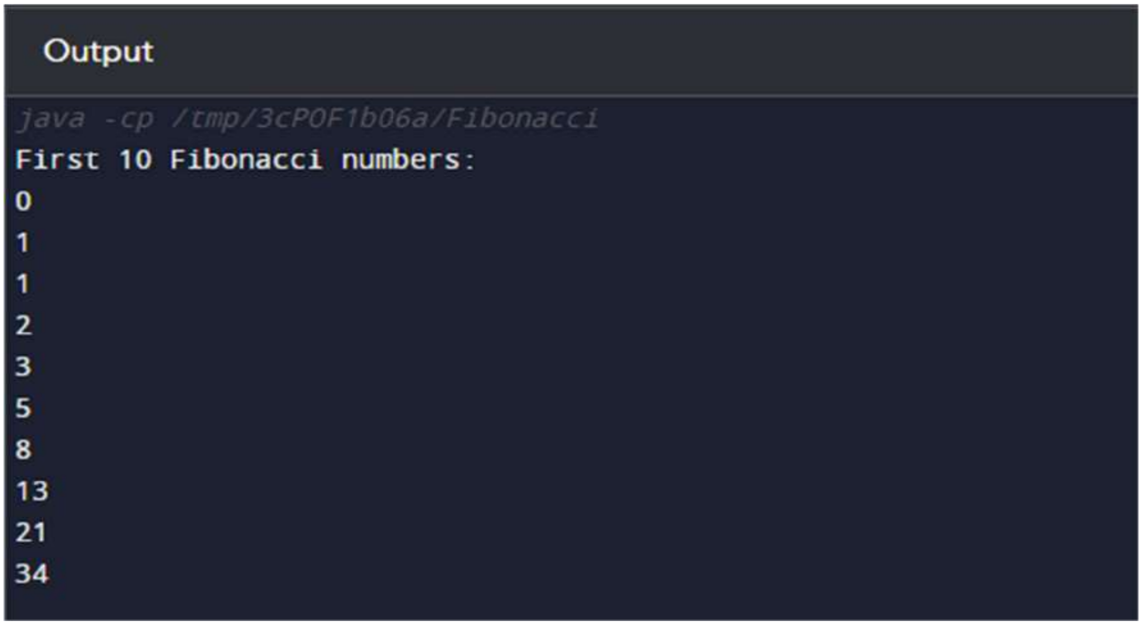


```
java -cp /tmp/eX8ixU1ldn/Main
Original array:
1 1 2 2 3 3 4 4 5 5

Unique elements:
1
2
3
4
5
```

Ques6: WAP to calculate first n Fibonacci numbers and store in an array.

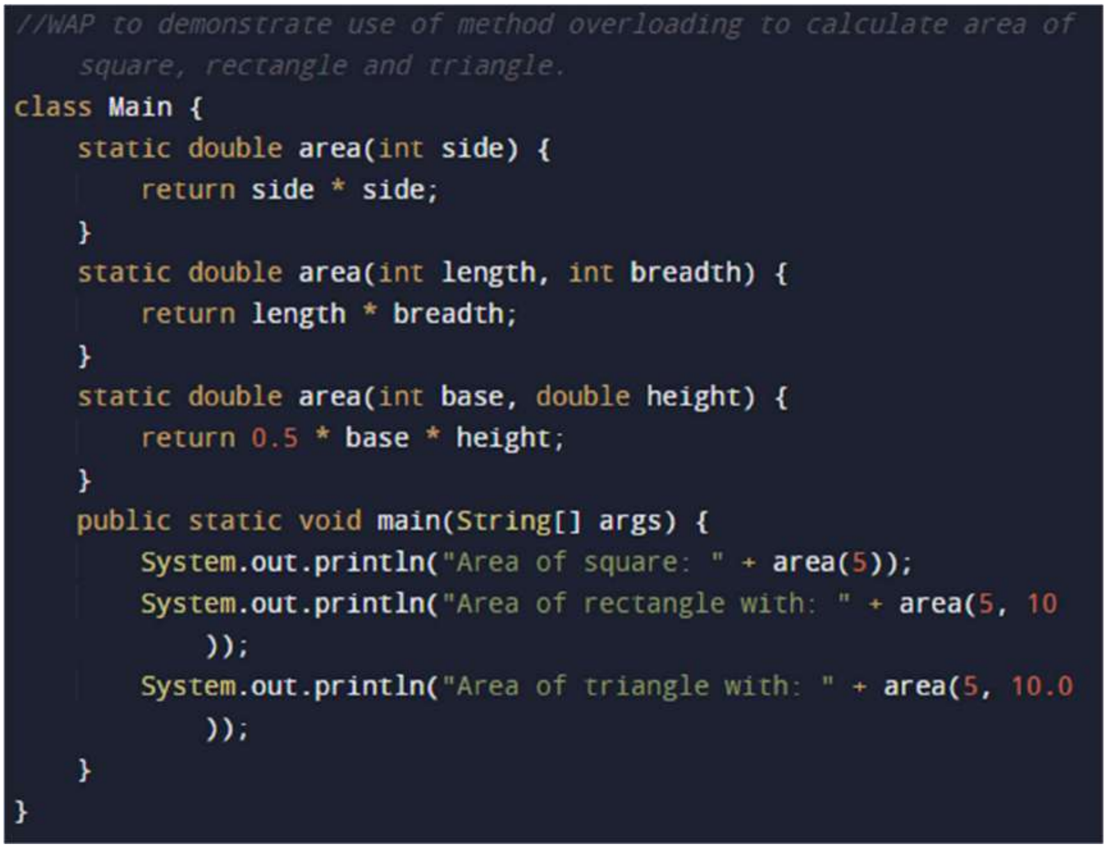


```java
//WAP to calculate first n Fibonacci numbers and store in an array.
public class Fibonacci {
    public static void main(String[] args) {
        int n = 10;
        int[] fibArray = new int[n];
        fibArray[0] = 0;
        fibArray[1] = 1;
        for (int i = 2; i < n; i++) {
            fibArray[i] = fibArray[i - 1] + fibArray[i - 2];
        }
        System.out.println("First " + n + " Fibonacci numbers:");
        for (int num : fibArray) {
            System.out.println(num);
        }
    }
}
```

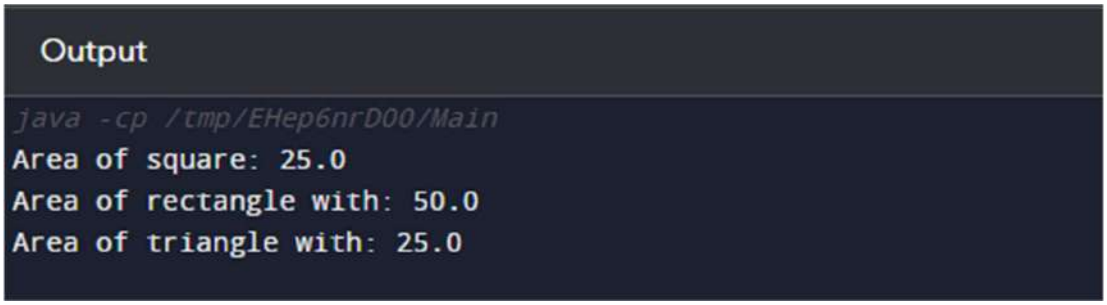


```
java -cp /tmp/3cP0F1b06a/Fibonacci
First 10 Fibonacci numbers:
0
1
1
2
3
5
8
13
21
34
```

Ques7: WAP to demonstrate use of method overloading to calculate area of square, rectangle and triangle.



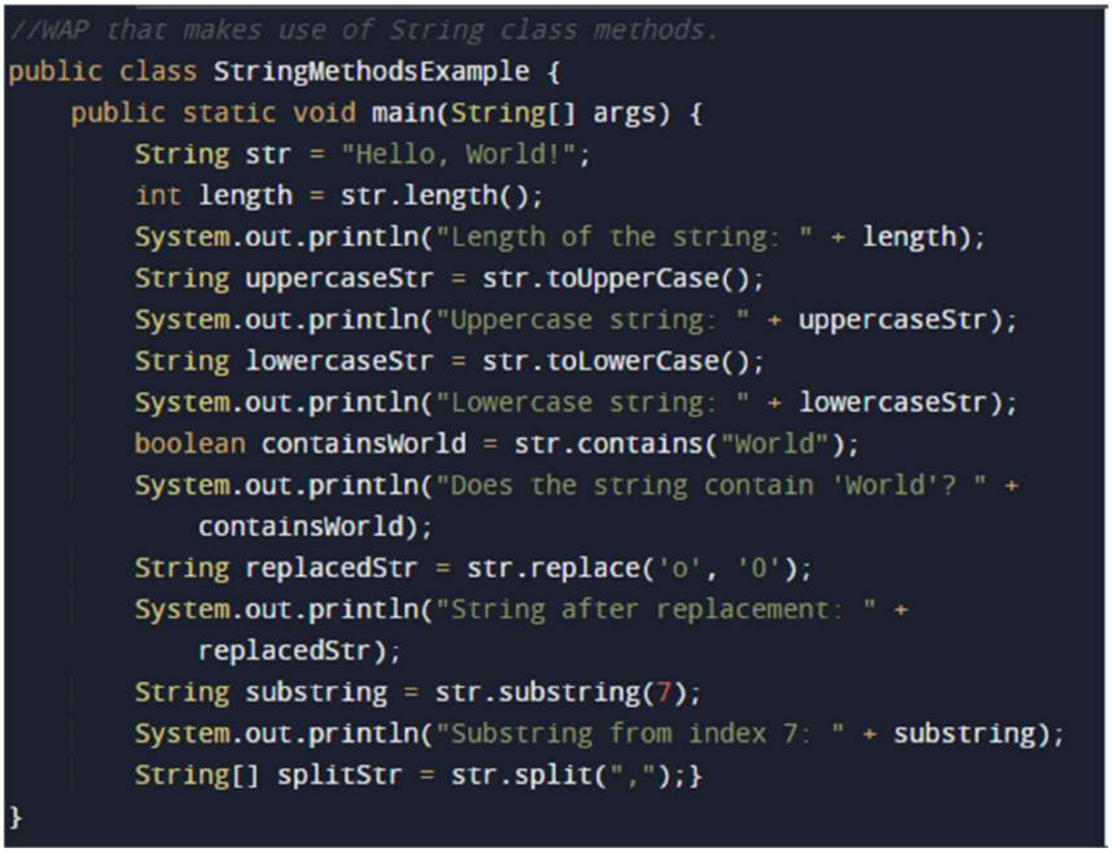
```
//WAP to demonstrate use of method overloading to calculate area of
    square, rectangle and triangle.
class Main {
    static double area(int side) {
        return side * side;
    }
    static double area(int length, int breadth) {
        return length * breadth;
    }
    static double area(int base, double height) {
        return 0.5 * base * height;
    }
    public static void main(String[] args) {
        System.out.println("Area of square: " + area(5));
        System.out.println("Area of rectangle with: " + area(5, 10
            ));
        System.out.println("Area of triangle with: " + area(5, 10.0
            ));
    }
}
```



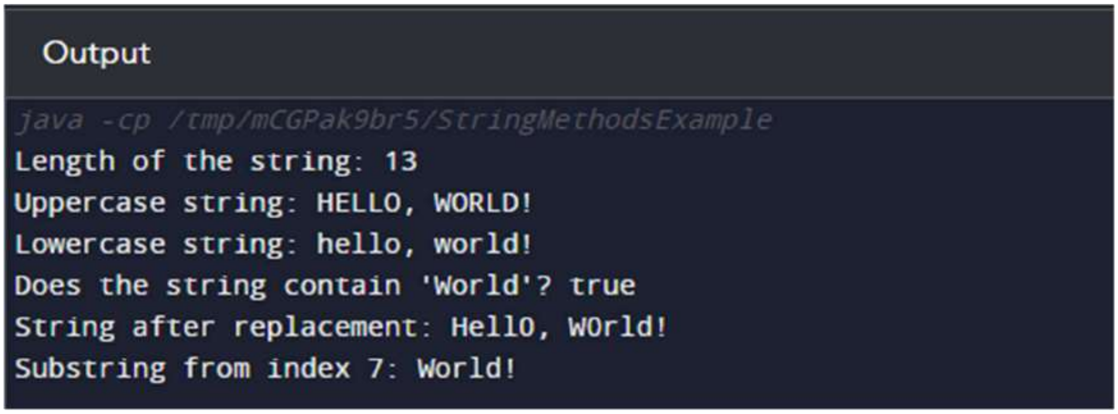
Output

```
java -cp /tmp/EHep6nrD0O/Main
Area of square: 25.0
Area of rectangle with: 50.0
Area of triangle with: 25.0
```

Ques8: WAP that makes use of String class methods.

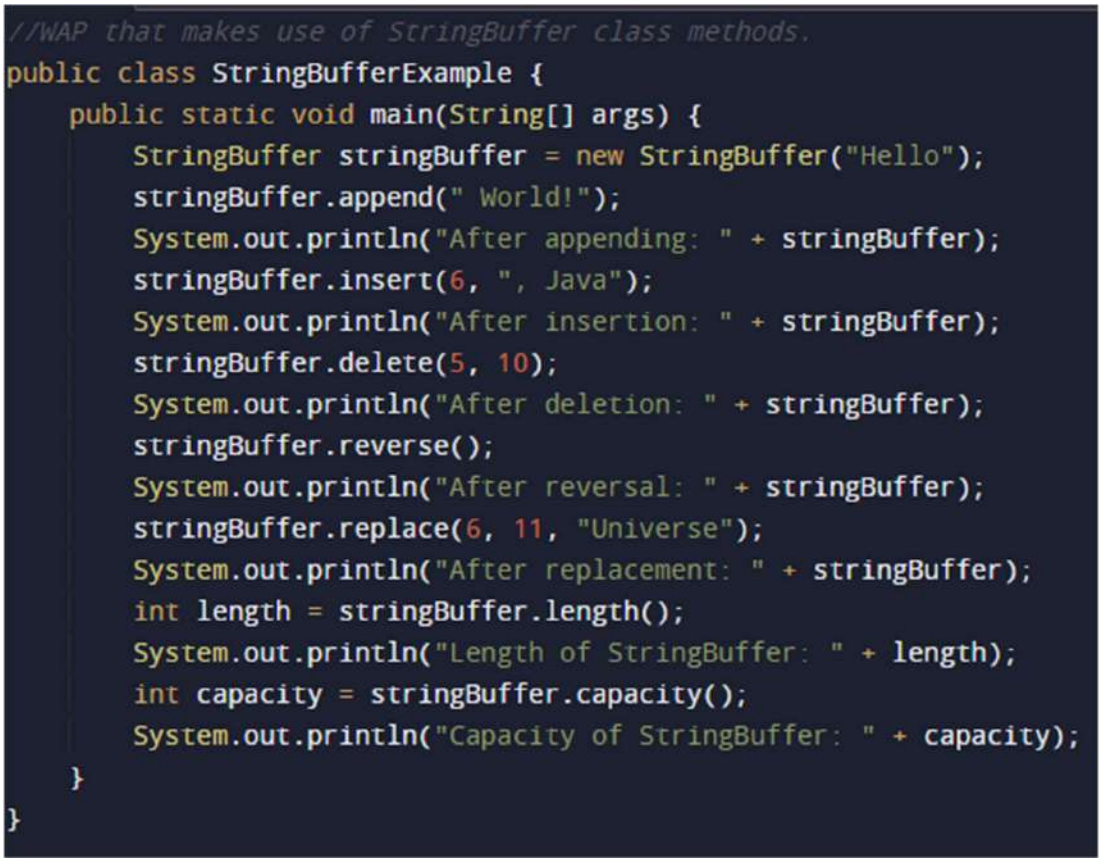


```java
//WAP that makes use of String class methods.
public class StringMethodsExample {
    public static void main(String[] args) {
        String str = "Hello, World!";
        int length = str.length();
        System.out.println("Length of the string: " + length);
        String uppercaseStr = str.toUpperCase();
        System.out.println("Uppercase string: " + uppercaseStr);
        String lowercaseStr = str.toLowerCase();
        System.out.println("Lowercase string: " + lowercaseStr);
        boolean containsWorld = str.contains("World");
        System.out.println("Does the string contain 'World'? " +
            containsWorld);
        String replacedStr = str.replace('o', 'O');
        System.out.println("String after replacement: " +
            replacedStr);
        String substring = str.substring(7);
        System.out.println("Substring from index 7: " + substring);
        String[] splitStr = str.split(",");}
}
```

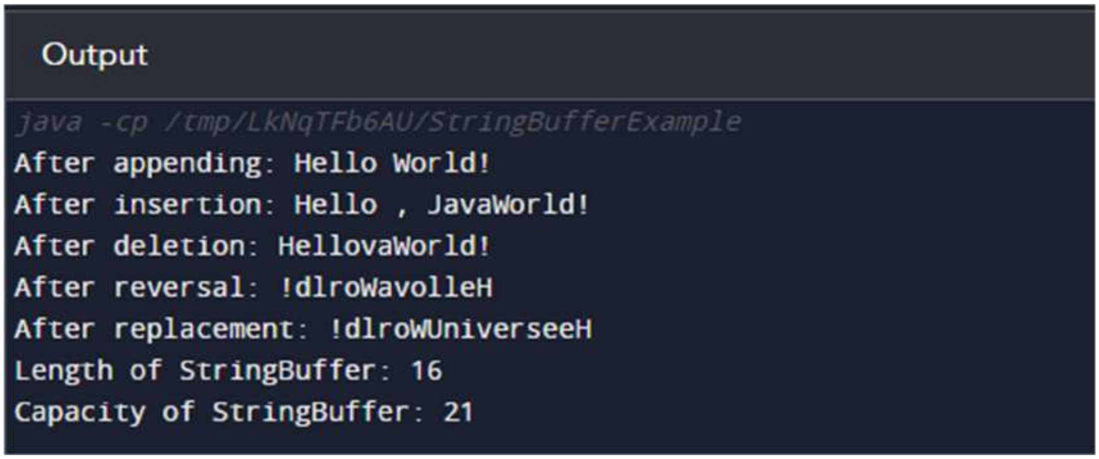


```
java -cp /tmp/mCGPak9br5/StringMethodsExample
Length of the string: 13
Uppercase string: HELLO, WORLD!
Lowercase string: hello, world!
Does the string contain 'World'? true
String after replacement: HellO, WOrld!
Substring from index 7: World!
```

Ques9: WAP that makes use of StringBuffer class methods.



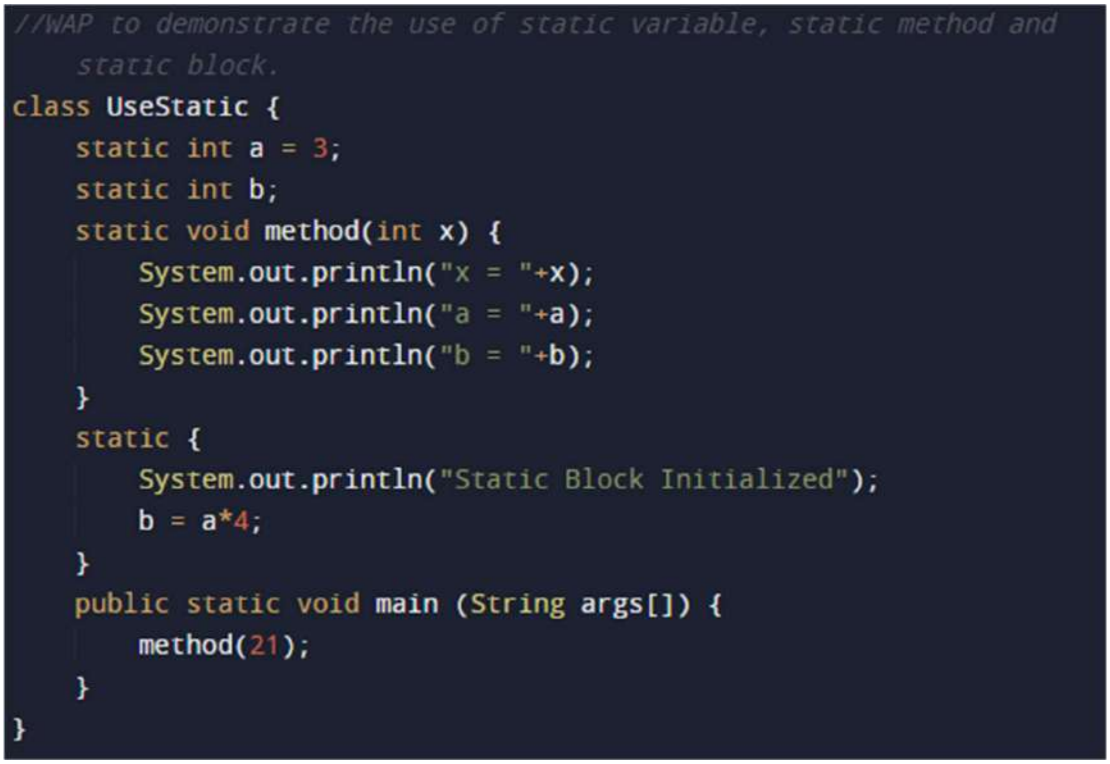
```
//WAP that makes use of StringBuffer class methods.
public class StringBufferExample {
    public static void main(String[] args) {
        StringBuffer stringBuffer = new StringBuffer("Hello");
        stringBuffer.append(" World!");
        System.out.println("After appending: " + stringBuffer);
        stringBuffer.insert(6, ", Java");
        System.out.println("After insertion: " + stringBuffer);
        stringBuffer.delete(5, 10);
        System.out.println("After deletion: " + stringBuffer);
        stringBuffer.reverse();
        System.out.println("After reversal: " + stringBuffer);
        stringBuffer.replace(6, 11, "Universe");
        System.out.println("After replacement: " + stringBuffer);
        int length = stringBuffer.length();
        System.out.println("Length of StringBuffer: " + length);
        int capacity = stringBuffer.capacity();
        System.out.println("Capacity of StringBuffer: " + capacity);
    }
}
```



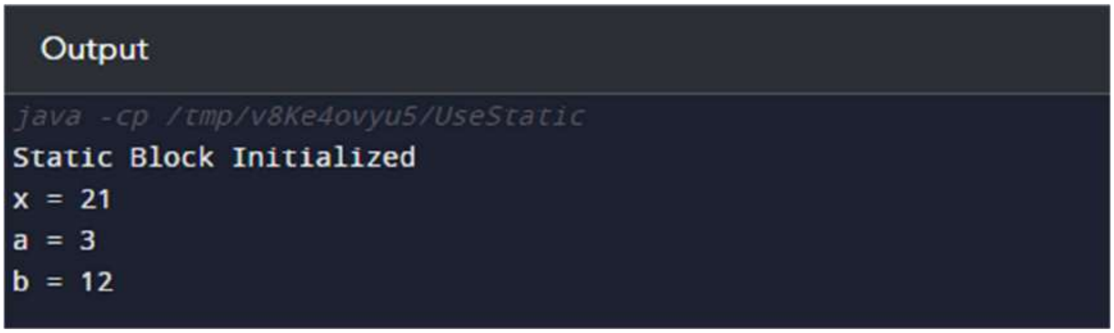
Output

```
java -cp /tmp/LkNqTFb6AU/StringBufferExample
After appending: Hello World!
After insertion: Hello , JavaWorld!
After deletion: HellovaWorld!
After reversal: !dlroWavolleH
After replacement: !dlroWUniverseeH
Length of StringBuffer: 16
Capacity of StringBuffer: 21
```

Ques10: WAP to demonstrate the use of static variable, static method and static block.

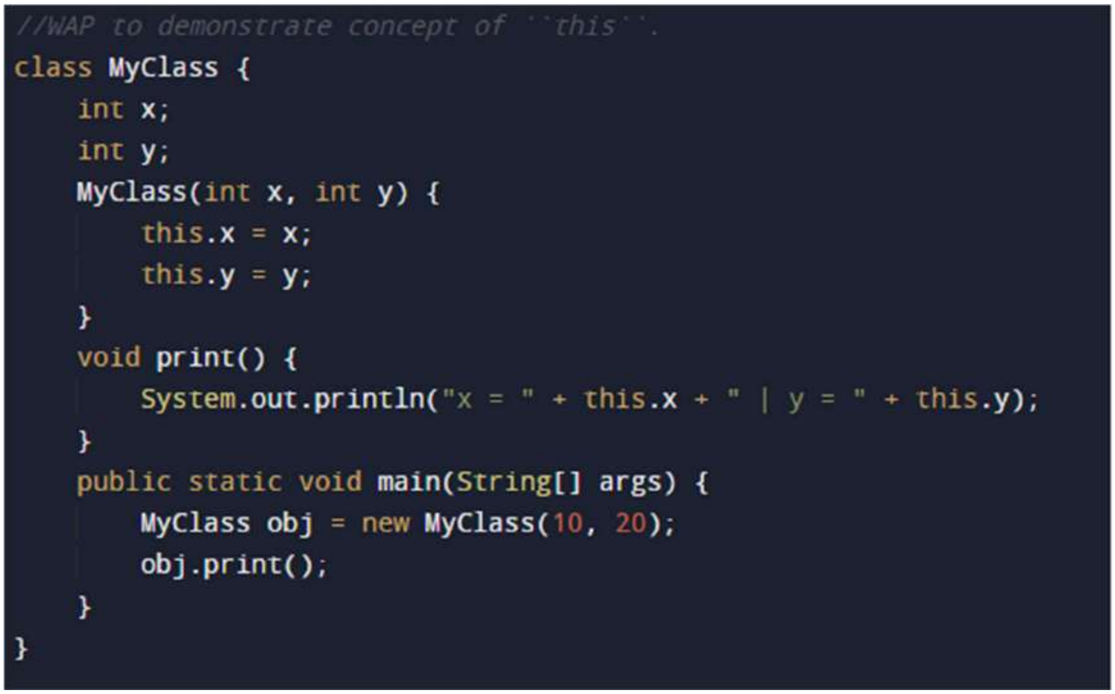


```java
//WAP to demonstrate the use of static variable, static method and
    static block.
class UseStatic {
    static int a = 3;
    static int b;
    static void method(int x) {
        System.out.println("x = "+x);
        System.out.println("a = "+a);
        System.out.println("b = "+b);
    }
    static {
        System.out.println("Static Block Initialized");
        b = a*4;
    }
    public static void main (String args[]) {
        method(21);
    }
}
```

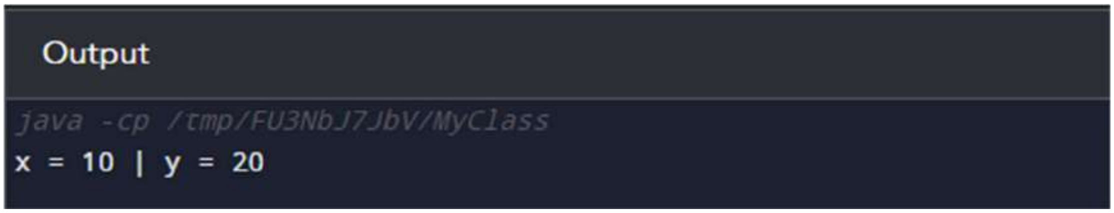


```
java -cp /tmp/v8Ke4ovyu5/UseStatic
Static Block Initialized
x = 21
a = 3
b = 12
```

Ques11: WAP to demonstrate concept of ``this``.

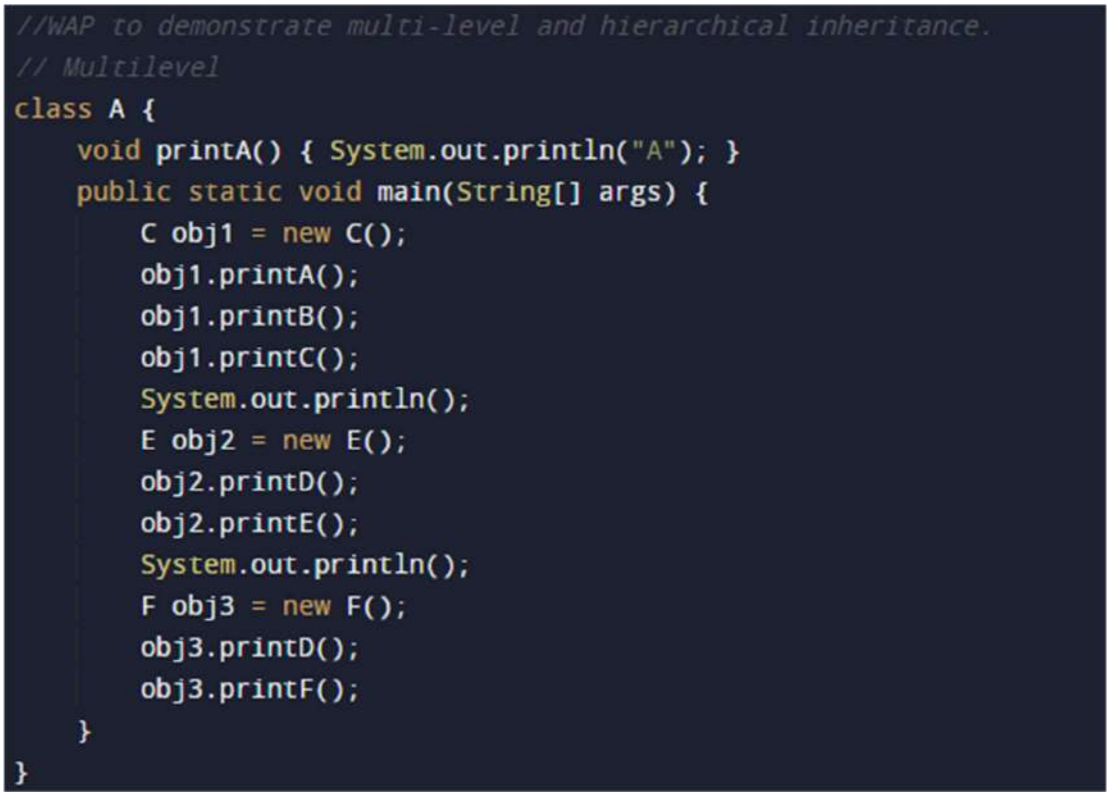


```
//WAP to demonstrate concept of ``this``.
class MyClass {
    int x;
    int y;
    MyClass(int x, int y) {
        this.x = x;
        this.y = y;
    }
    void print() {
        System.out.println("x = " + this.x + " | y = " + this.y);
    }
    public static void main(String[] args) {
        MyClass obj = new MyClass(10, 20);
        obj.print();
    }
}
```

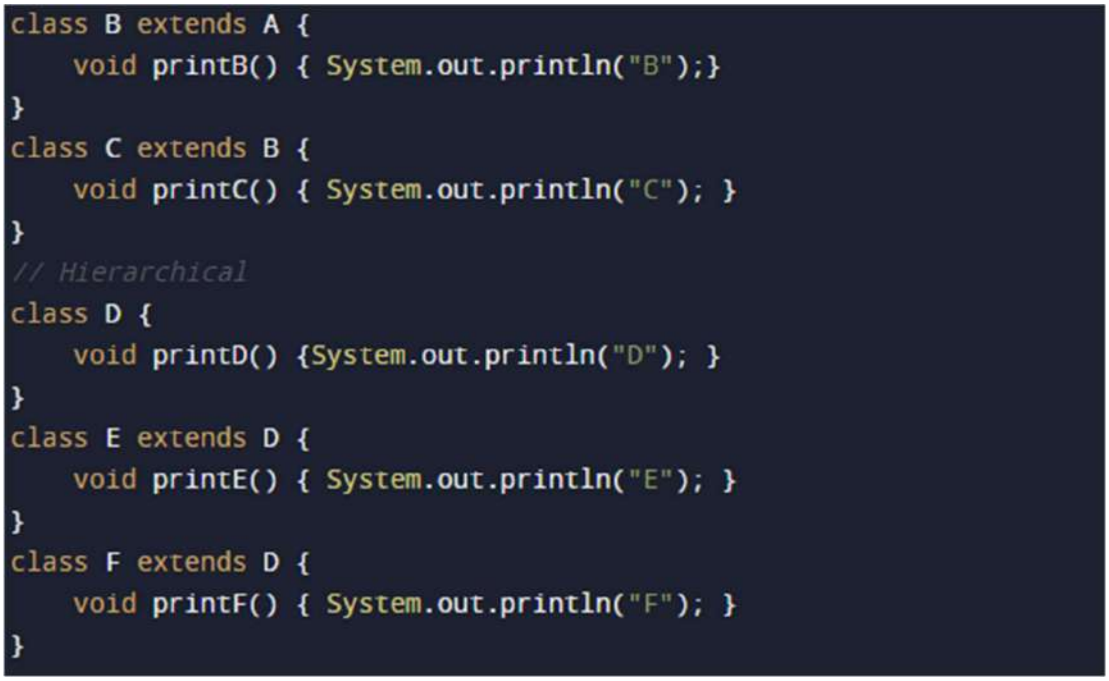


```
java -cp /tmp/FU3NbJ7JbV/MyClass
x = 10 | y = 20
```

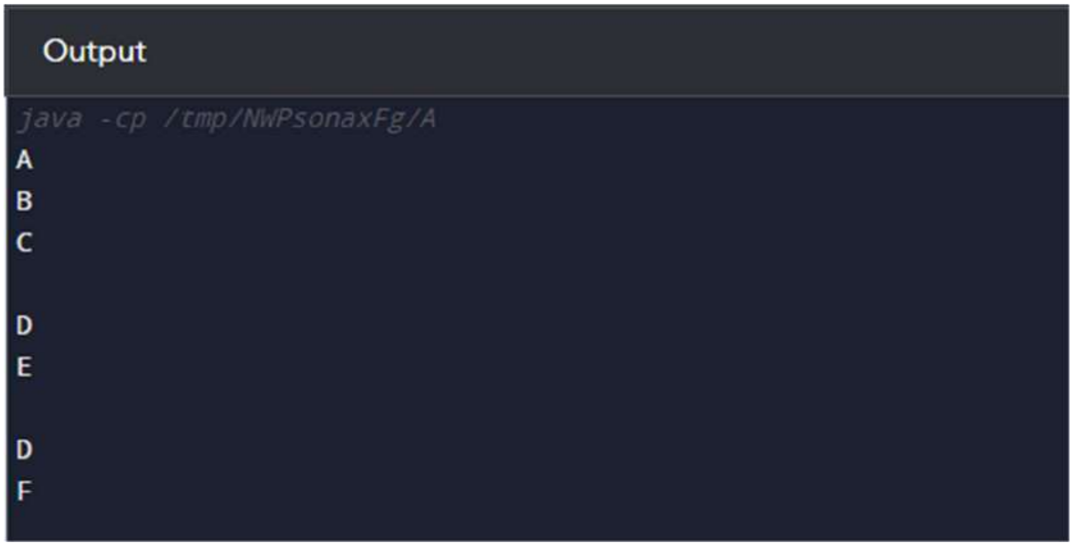
Ques12: WAP to demonstrate multi-level and hierarchical inheritance.



```java
//WAP to demonstrate multi-level and hierarchical inheritance.
// Multilevel
class A {
    void printA() { System.out.println("A"); }
    public static void main(String[] args) {
        C obj1 = new C();
        obj1.printA();
        obj1.printB();
        obj1.printC();
        System.out.println();
        E obj2 = new E();
        obj2.printD();
        obj2.printE();
        System.out.println();
        F obj3 = new F();
        obj3.printD();
        obj3.printF();
    }
}
```

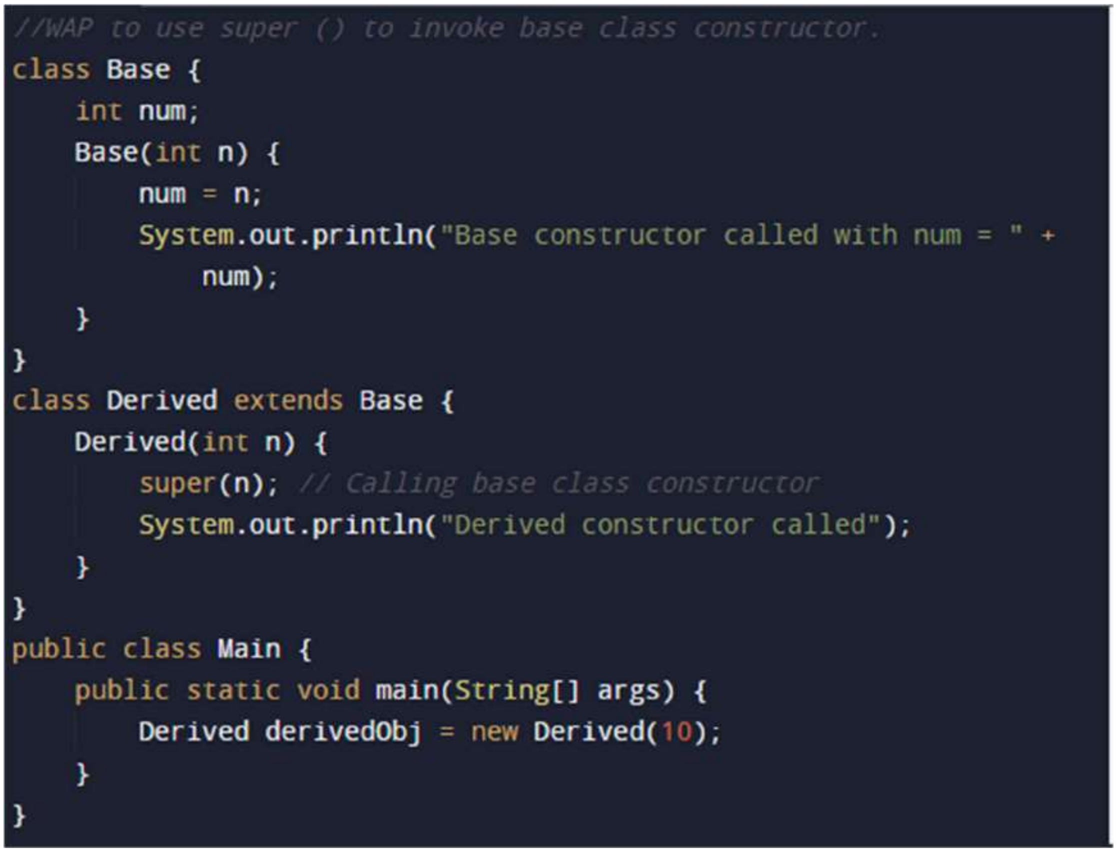


```java
class B extends A {
    void printB() { System.out.println("B");}
}
class C extends B {
    void printC() { System.out.println("C"); }
}
// Hierarchical
class D {
    void printD() {System.out.println("D"); }
}
class E extends D {
    void printE() { System.out.println("E"); }
}
class F extends D {
    void printF() { System.out.println("F"); }
}
```

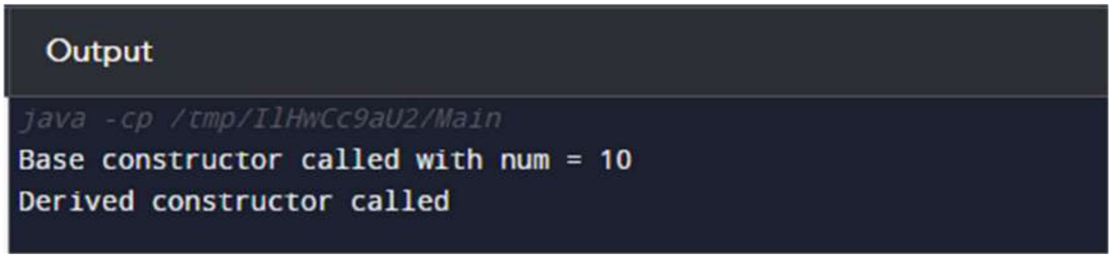


```
java -cp /tmp/NWPsonaxFg/A
A
B
C

D
E

D
F
```

Ques13: WAP to use super () to invoke base class constructor.

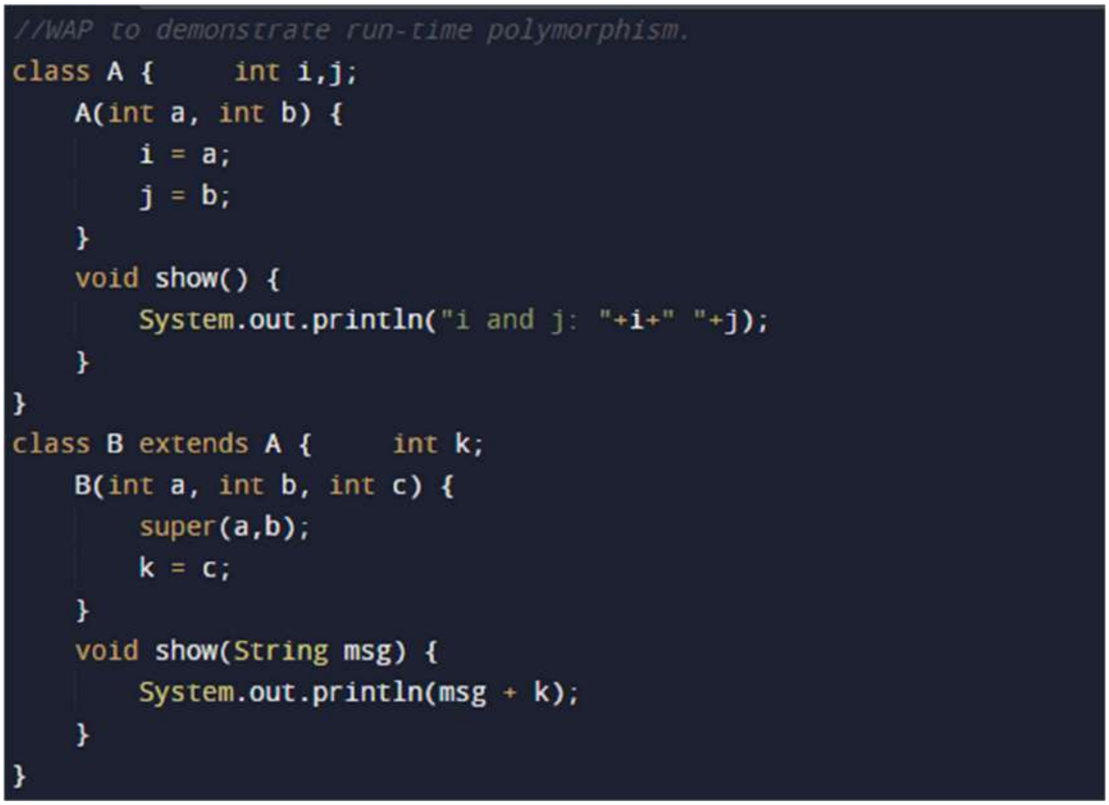


```java
//WAP to use super () to invoke base class constructor.
class Base {
    int num;
    Base(int n) {
        num = n;
        System.out.println("Base constructor called with num = " +
            num);
    }
}
class Derived extends Base {
    Derived(int n) {
        super(n); // Calling base class constructor
        System.out.println("Derived constructor called");
    }
}
public class Main {
    public static void main(String[] args) {
        Derived derivedObj = new Derived(10);
    }
}
```

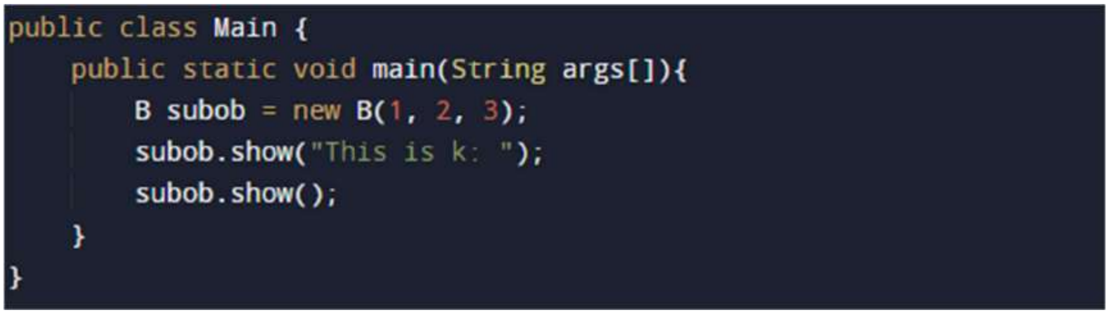


```
java -cp /tmp/IlHwCc9aU2/Main
Base constructor called with num = 10
Derived constructor called
```

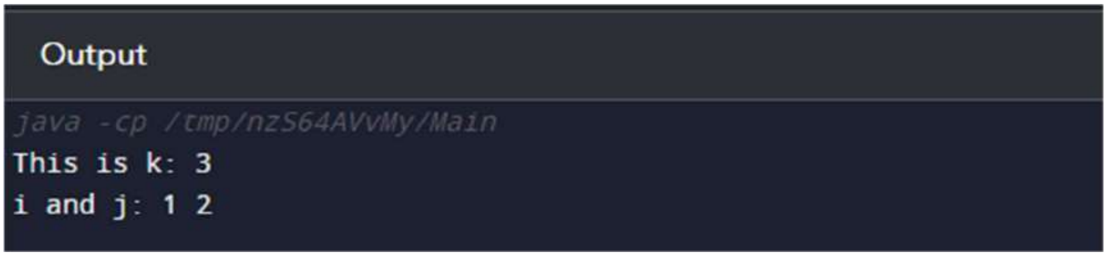
Ques14: WAP to demonstrate run-time polymorphism.



```java
//WAP to demonstrate run-time polymorphism.
class A {     int i,j;
    A(int a, int b) {
        i = a;
        j = b;
    }
    void show() {
        System.out.println("i and j: "+i+" "+j);
    }
}
class B extends A {     int k;
    B(int a, int b, int c) {
        super(a,b);
        k = c;
    }
    void show(String msg) {
        System.out.println(msg + k);
    }
}
```



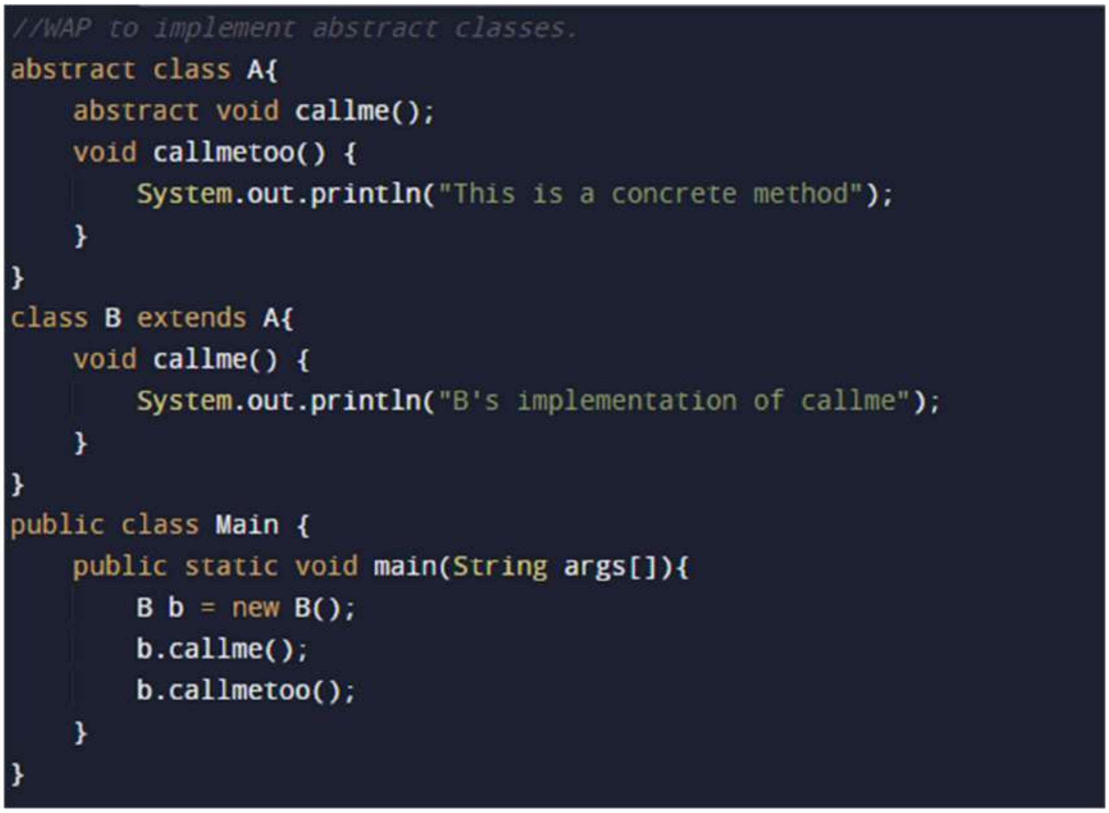
```java
public class Main {
    public static void main(String args[]){
        B subob = new B(1, 2, 3);
        subob.show("This is k: ");
        subob.show();
    }
}
```



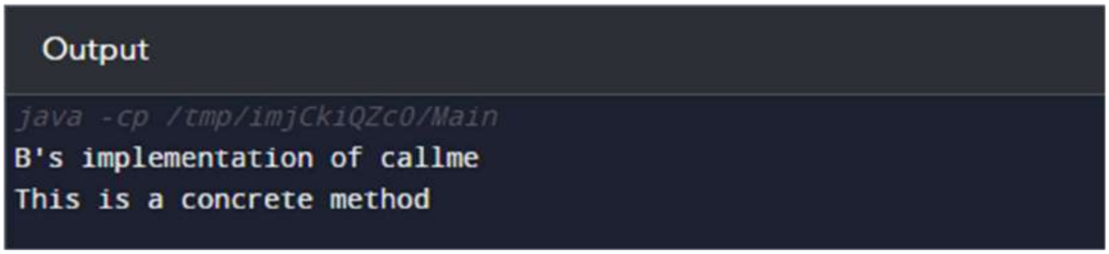
Output

```
java -cp /tmp/nzS64AVvMy/Main
This is k: 3
i and j: 1 2
```

Ques15: WAP to implement abstract classes.

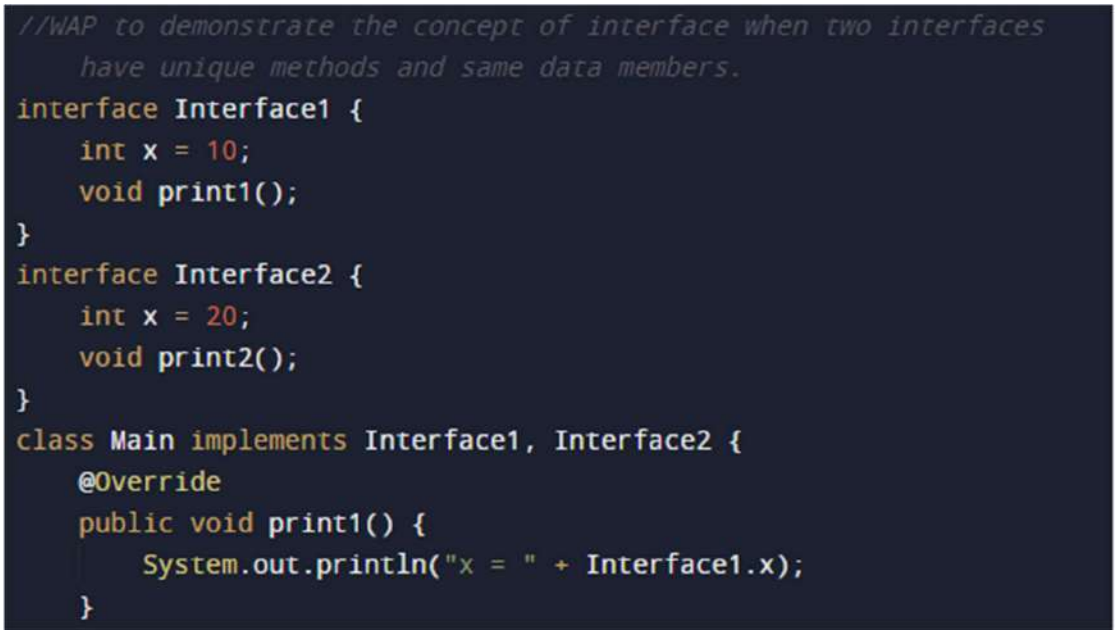


```java
//WAP to implement abstract classes.
abstract class A{
    abstract void callme();
    void callmetoo() {
        System.out.println("This is a concrete method");
    }
}
class B extends A{
    void callme() {
        System.out.println("B's implementation of callme");
    }
}
public class Main {
    public static void main(String args[]){
        B b = new B();
        b.callme();
        b.callmetoo();
    }
}
```

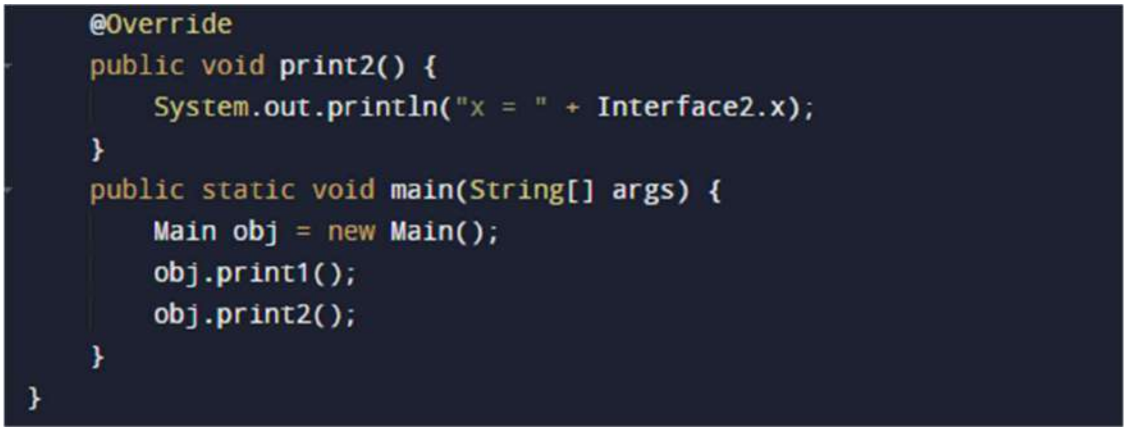


```
java -cp /tmp/imjCkiQZc0/Main
B's implementation of callme
This is a concrete method
```

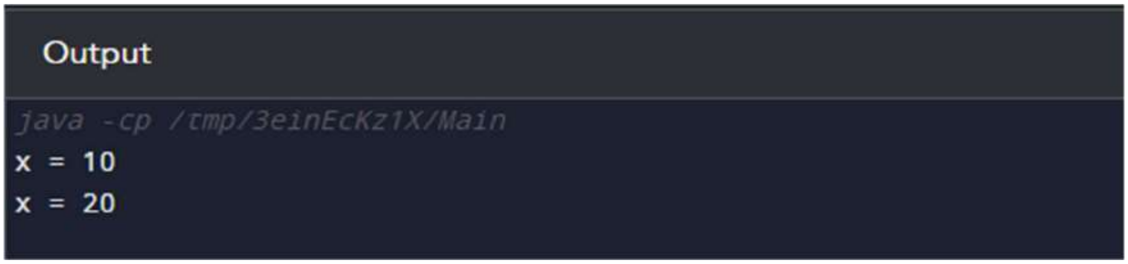
Ques16: WAP to demonstrate the concept of interface when two interfaces have unique methods and same data members.



```java
//WAP to demonstrate the concept of interface when two interfaces
    have unique methods and same data members.
interface Interface1 {
    int x = 10;
    void print1();
}
interface Interface2 {
    int x = 20;
    void print2();
}
class Main implements Interface1, Interface2 {
    @Override
    public void print1() {
        System.out.println("x = " + Interface1.x);
    }
```

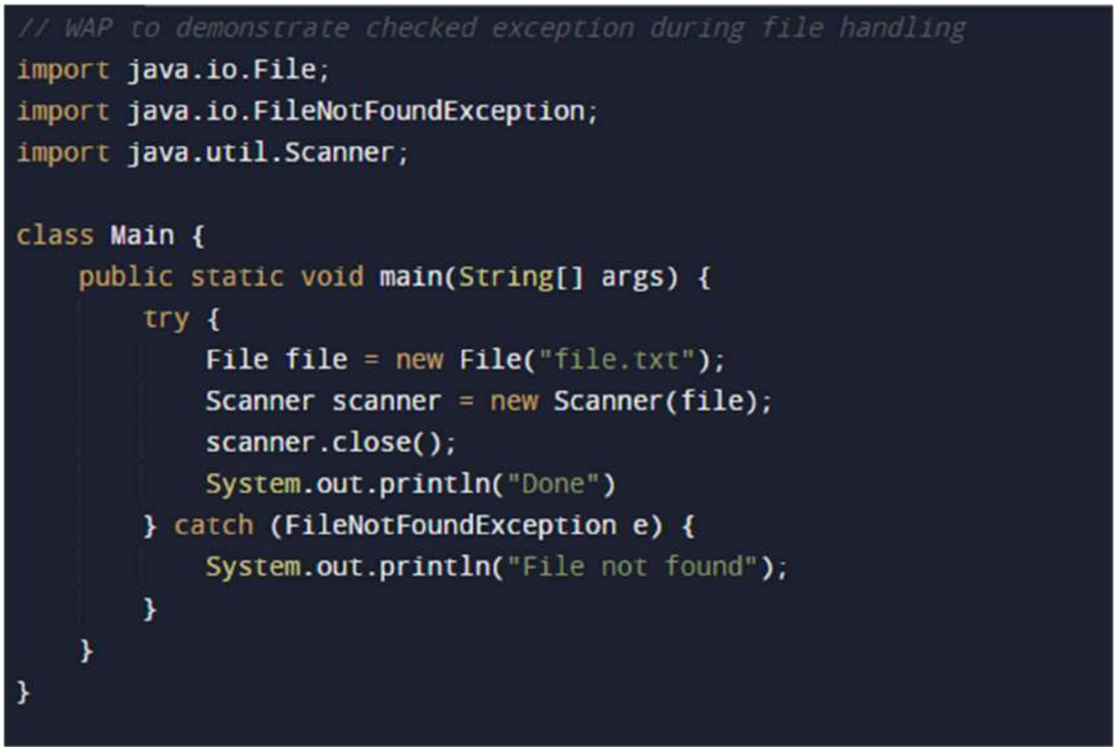


```java
    @Override
    public void print2() {
        System.out.println("x = " + Interface2.x);
    }
    public static void main(String[] args) {
        Main obj = new Main();
        obj.print1();
        obj.print2();
    }
}
```

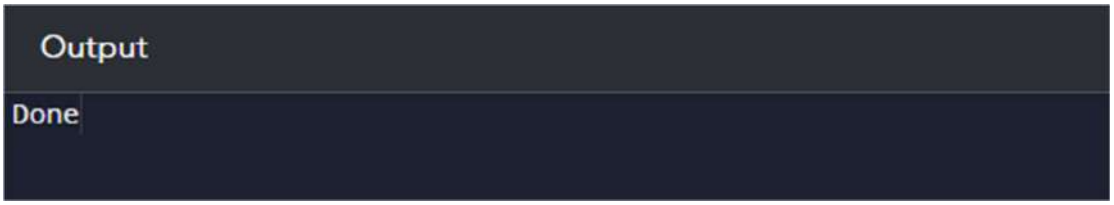


```
java -cp /tmp/3einEcKz1X/Main
x = 10
x = 20
```

Ques17: WAP to demonstrate checked exception during file handling.

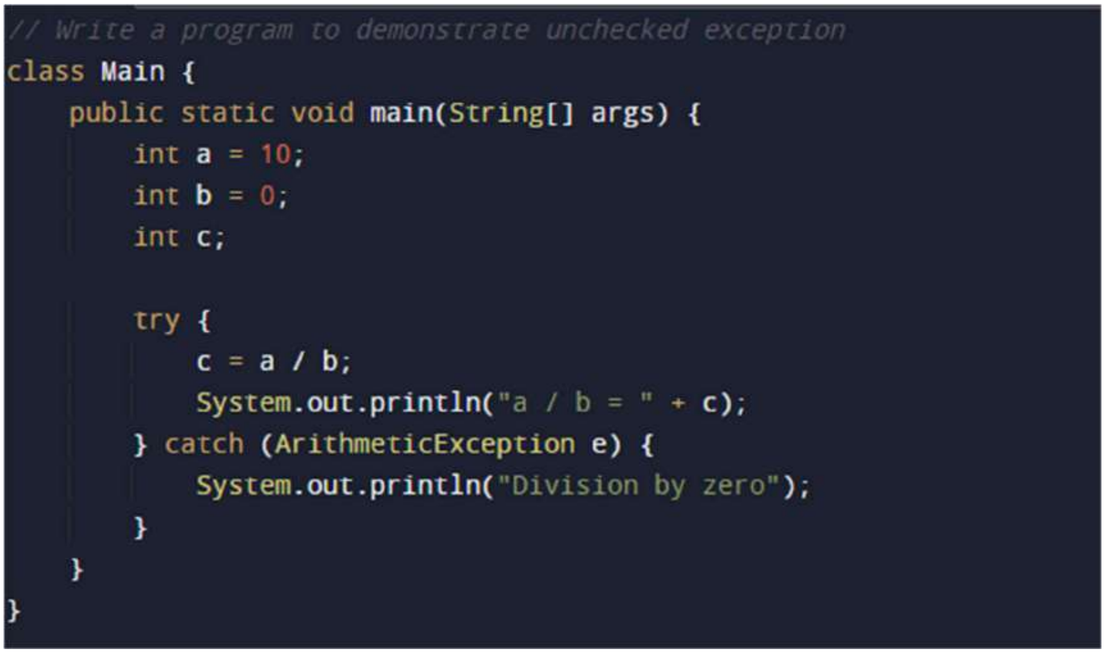


```
// WAP to demonstrate checked exception during file handling
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        try {
            File file = new File("file.txt");
            Scanner scanner = new Scanner(file);
            scanner.close();
            System.out.println("Done")
        } catch (FileNotFoundException e) {
            System.out.println("File not found");
        }
    }
}
```

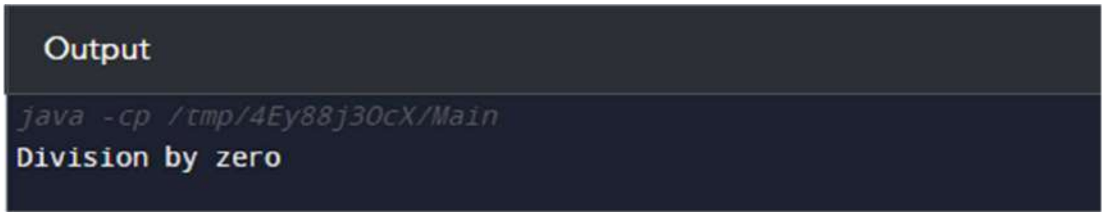


```
Done
```

Ques18: Write a program to demonstrate unchecked exception.



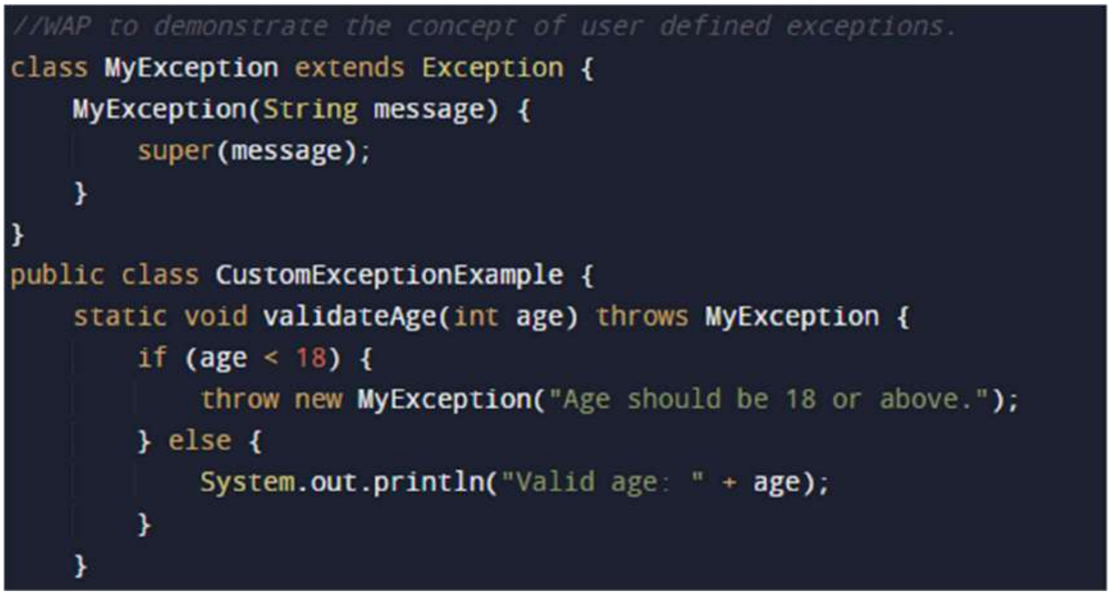
```java
// Write a program to demonstrate unchecked exception
class Main {
    public static void main(String[] args) {
        int a = 10;
        int b = 0;
        int c;

        try {
            c = a / b;
            System.out.println("a / b = " + c);
        } catch (ArithmeticException e) {
            System.out.println("Division by zero");
        }
    }
}
```



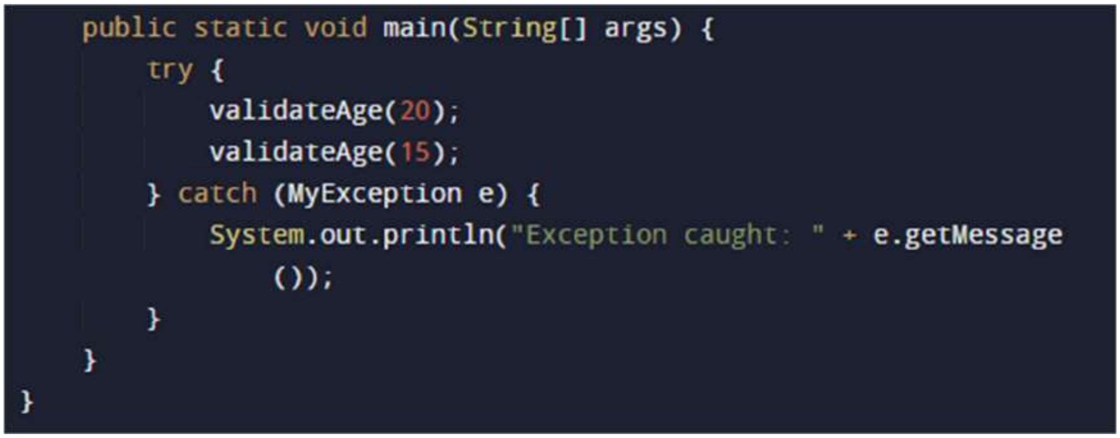
Output

```
java -cp /tmp/4Ey88j3OcX/Main
Division by zero
```

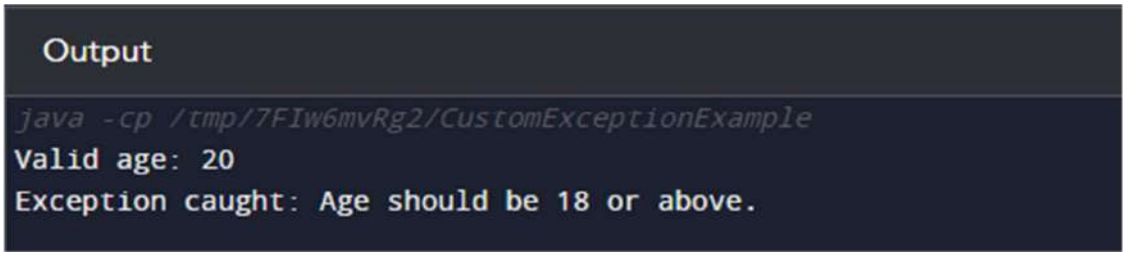
Ques19: WAP to demonstrate the concept of user defined exceptions.



```
//WAP to demonstrate the concept of user defined exceptions.
class MyException extends Exception {
    MyException(String message) {
        super(message);
    }
}
public class CustomExceptionExample {
    static void validateAge(int age) throws MyException {
        if (age < 18) {
            throw new MyException("Age should be 18 or above.");
        } else {
            System.out.println("Valid age: " + age);
        }
    }
```



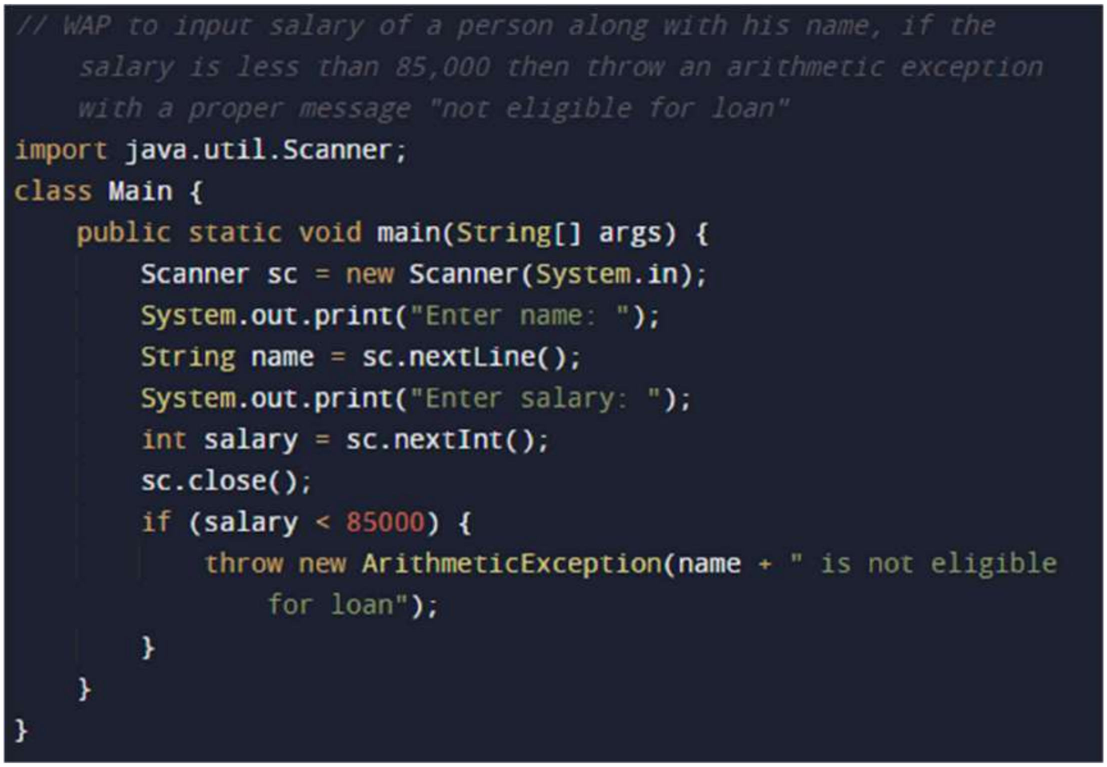
```
    public static void main(String[] args) {
        try {
            validateAge(20);
            validateAge(15);
        } catch (MyException e) {
            System.out.println("Exception caught: " + e.getMessage
                ());
        }
    }
}
```



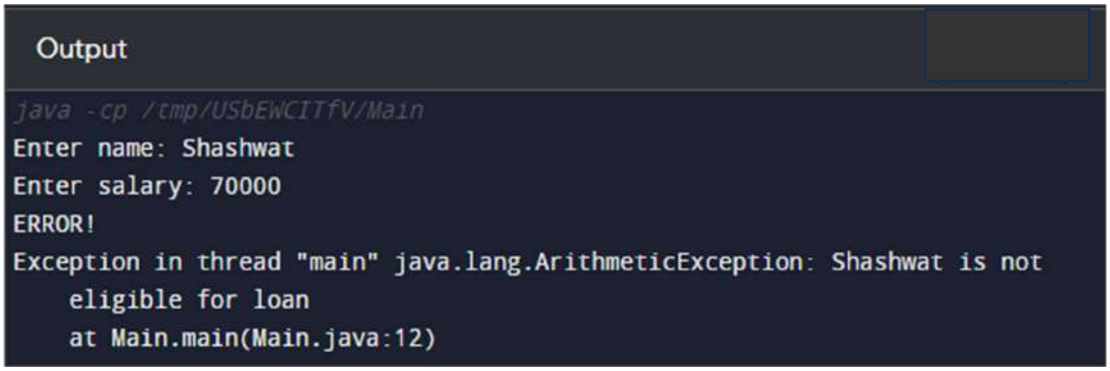
Output

```
java -cp /tmp/7FIw6mvRg2/CustomExceptionExample
Valid age: 20
Exception caught: Age should be 18 or above.
```

Ques20: WAP to input salary of a person along with his name, if the salary is less than 85,000 then throw an arithmetic exception with a proper message “not eligible for loan”.

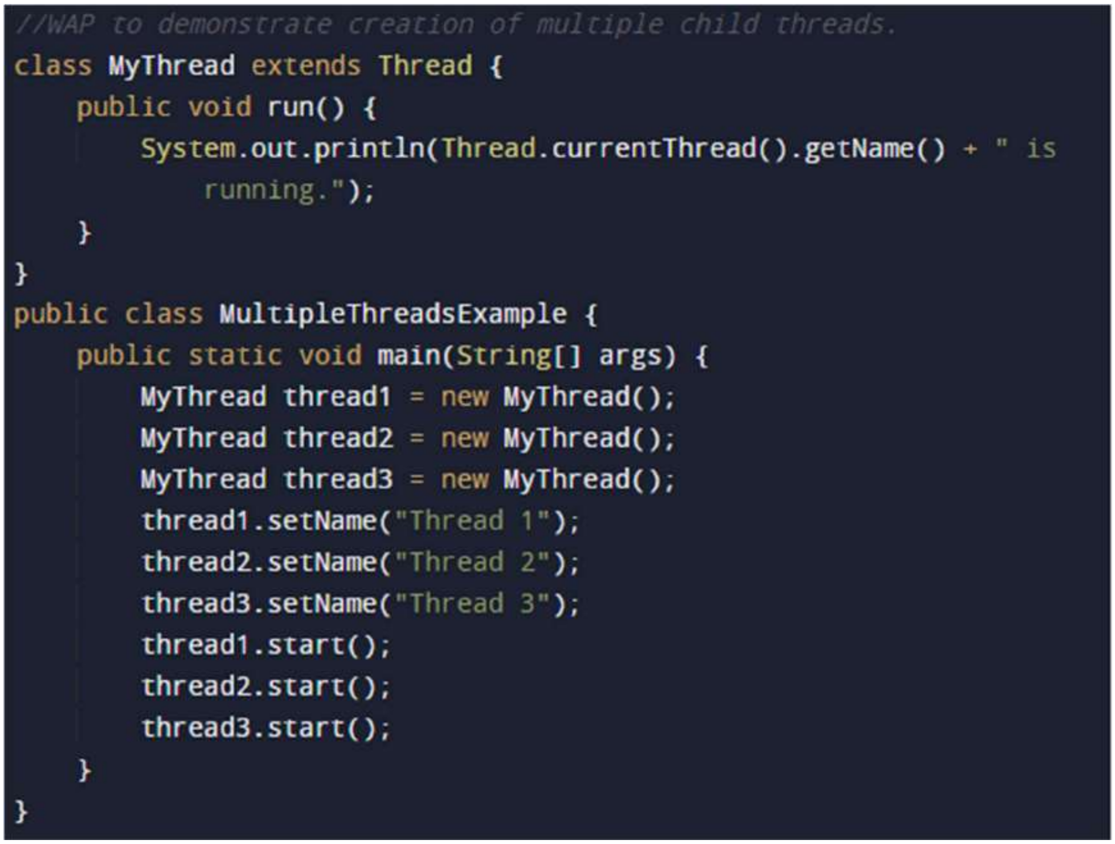


```java
// WAP to input salary of a person along with his name, if the
    salary is less than 85,000 then throw an arithmetic exception
    with a proper message "not eligible for loan"
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter name: ");
        String name = sc.nextLine();
        System.out.print("Enter salary: ");
        int salary = sc.nextInt();
        sc.close();
        if (salary < 85000) {
            throw new ArithmeticException(name + " is not eligible
                for loan");
        }
    }
}
```

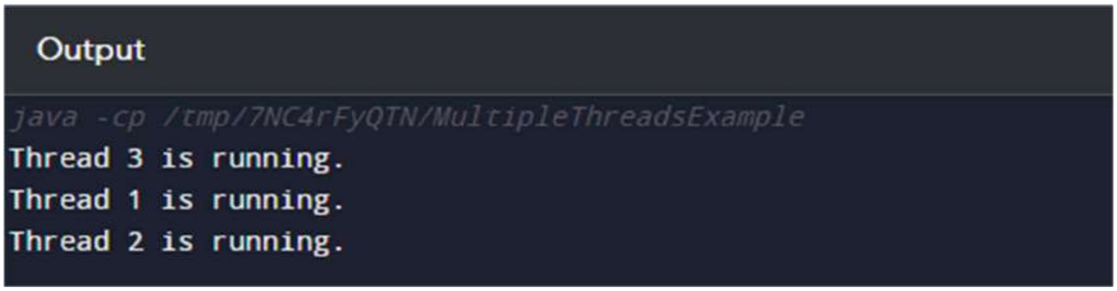


```
java -cp /tmp/USbEWCITfV/Main
Enter name: Shashwat
Enter salary: 70000
ERROR!
Exception in thread "main" java.lang.ArithmeticException: Shashwat is not
    eligible for loan
    at Main.main(Main.java:12)
```

Ques21: WAP to demonstrate creation of multiple child threads.



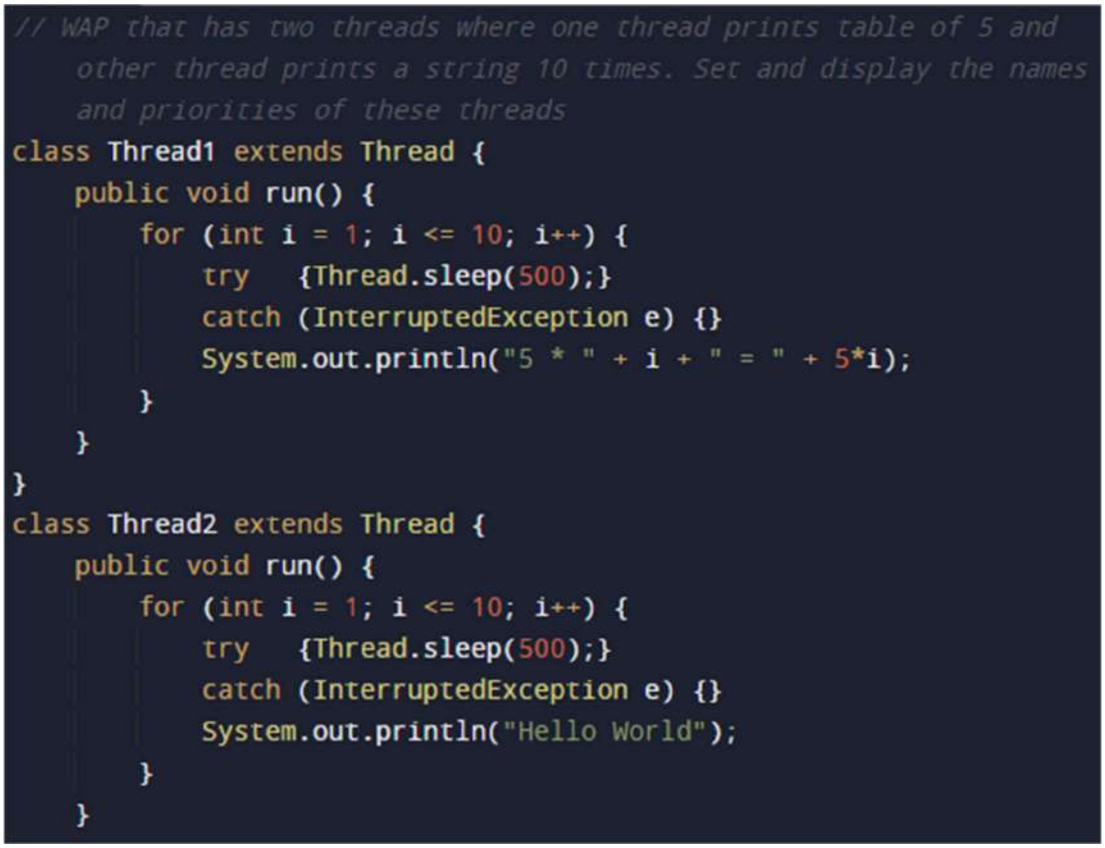
```java
//WAP to demonstrate creation of multiple child threads.
class MyThread extends Thread {
    public void run() {
        System.out.println(Thread.currentThread().getName() + " is
            running.");
    }
}
public class MultipleThreadsExample {
    public static void main(String[] args) {
        MyThread thread1 = new MyThread();
        MyThread thread2 = new MyThread();
        MyThread thread3 = new MyThread();
        thread1.setName("Thread 1");
        thread2.setName("Thread 2");
        thread3.setName("Thread 3");
        thread1.start();
        thread2.start();
        thread3.start();
    }
}
```



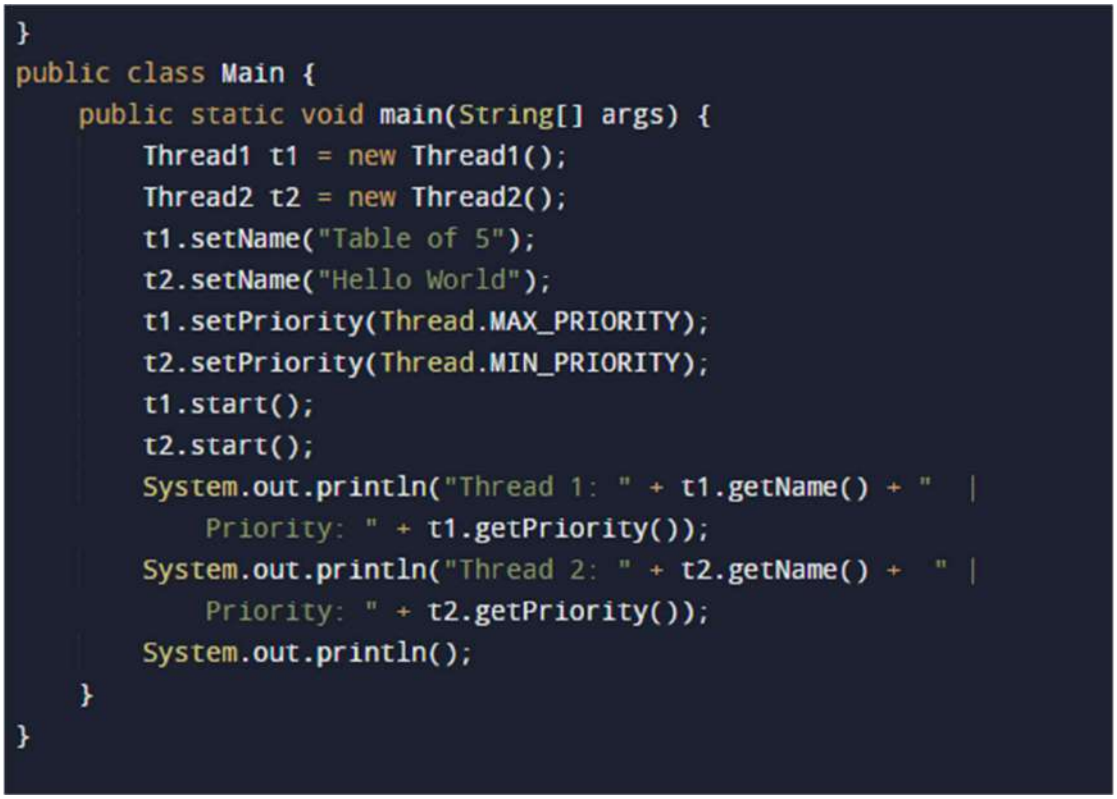
Output

```
java -cp /tmp/7NC4rFyQTN/MultipleThreadsExample
Thread 3 is running.
Thread 1 is running.
Thread 2 is running.
```

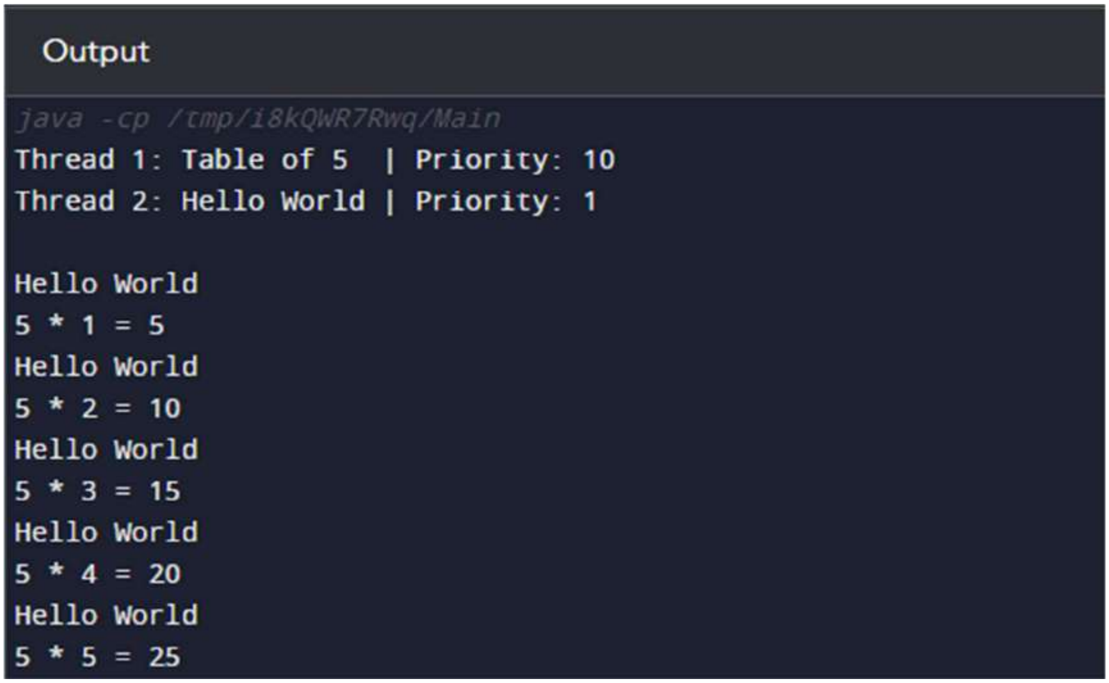
Ques22: WAP that has two threads where one thread prints table of 5 and other thread prints a string 10 times. Set and display the names and priorities of these threads.



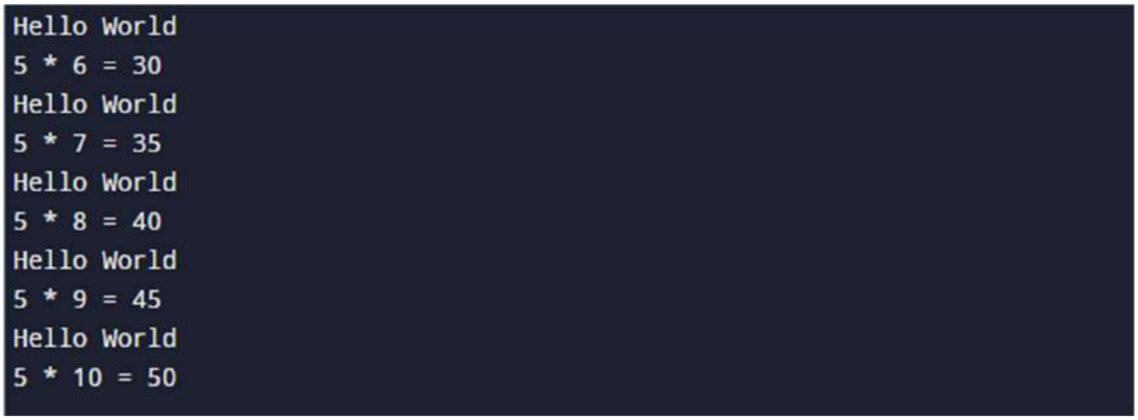
```java
// WAP that has two threads where one thread prints table of 5 and
    other thread prints a string 10 times. Set and display the names
    and priorities of these threads
class Thread1 extends Thread {
    public void run() {
        for (int i = 1; i <= 10; i++) {
            try   {Thread.sleep(500);}
            catch (InterruptedException e) {}
            System.out.println("5 * " + i + " = " + 5*i);
        }
    }
}
class Thread2 extends Thread {
    public void run() {
        for (int i = 1; i <= 10; i++) {
            try   {Thread.sleep(500);}
            catch (InterruptedException e) {}
            System.out.println("Hello World");
        }
    }
```



```java
}
public class Main {
    public static void main(String[] args) {
        Thread1 t1 = new Thread1();
        Thread2 t2 = new Thread2();
        t1.setName("Table of 5");
        t2.setName("Hello World");
        t1.setPriority(Thread.MAX_PRIORITY);
        t2.setPriority(Thread.MIN_PRIORITY);
        t1.start();
        t2.start();
        System.out.println("Thread 1: " + t1.getName() + "  |
            Priority: " + t1.getPriority());
        System.out.println("Thread 2: " + t2.getName() +  " |
            Priority: " + t2.getPriority());
        System.out.println();
    }
}
```

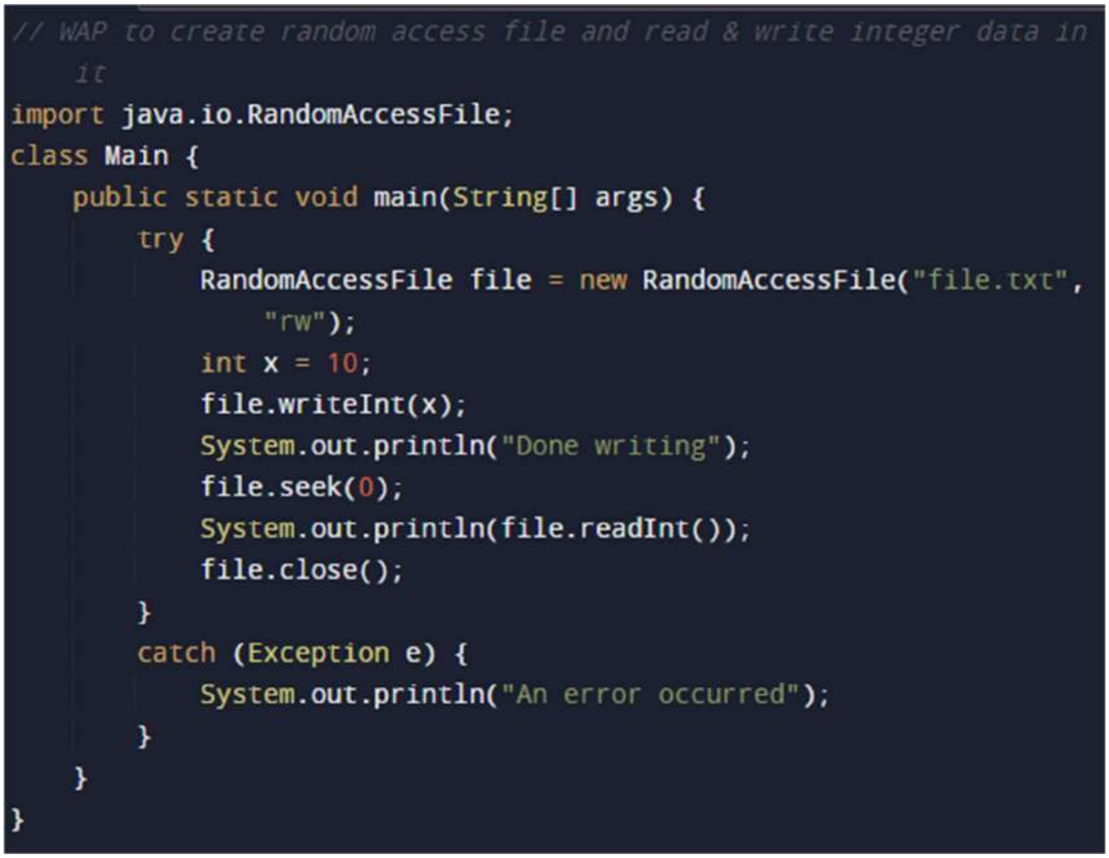


```
java -cp /tmp/i8kQWR7Rwq/Main
Thread 1: Table of 5  | Priority: 10
Thread 2: Hello World | Priority: 1

Hello World
5 * 1 = 5
Hello World
5 * 2 = 10
Hello World
5 * 3 = 15
Hello World
5 * 4 = 20
Hello World
5 * 5 = 25
```

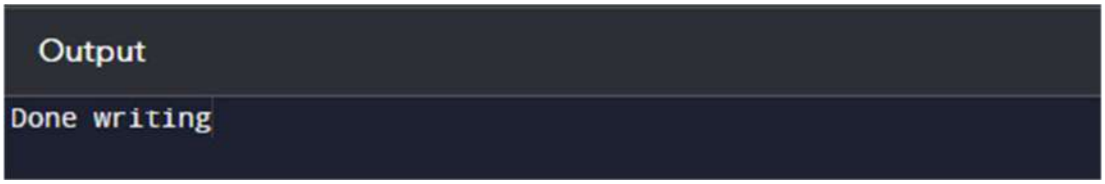


```
Hello World
5 * 6 = 30
Hello World
5 * 7 = 35
Hello World
5 * 8 = 40
Hello World
5 * 9 = 45
Hello World
5 * 10 = 50
```

Ques23: WAP to create random access file and read & write integer data in it.



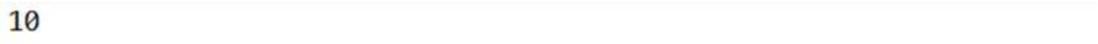
```
// WAP to create random access file and read & write integer data in
    it
import java.io.RandomAccessFile;
class Main {
    public static void main(String[] args) {
        try {
            RandomAccessFile file = new RandomAccessFile("file.txt",
                "rw");
            int x = 10;
            file.writeInt(x);
            System.out.println("Done writing");
            file.seek(0);
            System.out.println(file.readInt());
            file.close();
        }
        catch (Exception e) {
            System.out.println("An error occurred");
        }
    }
}
```



Output

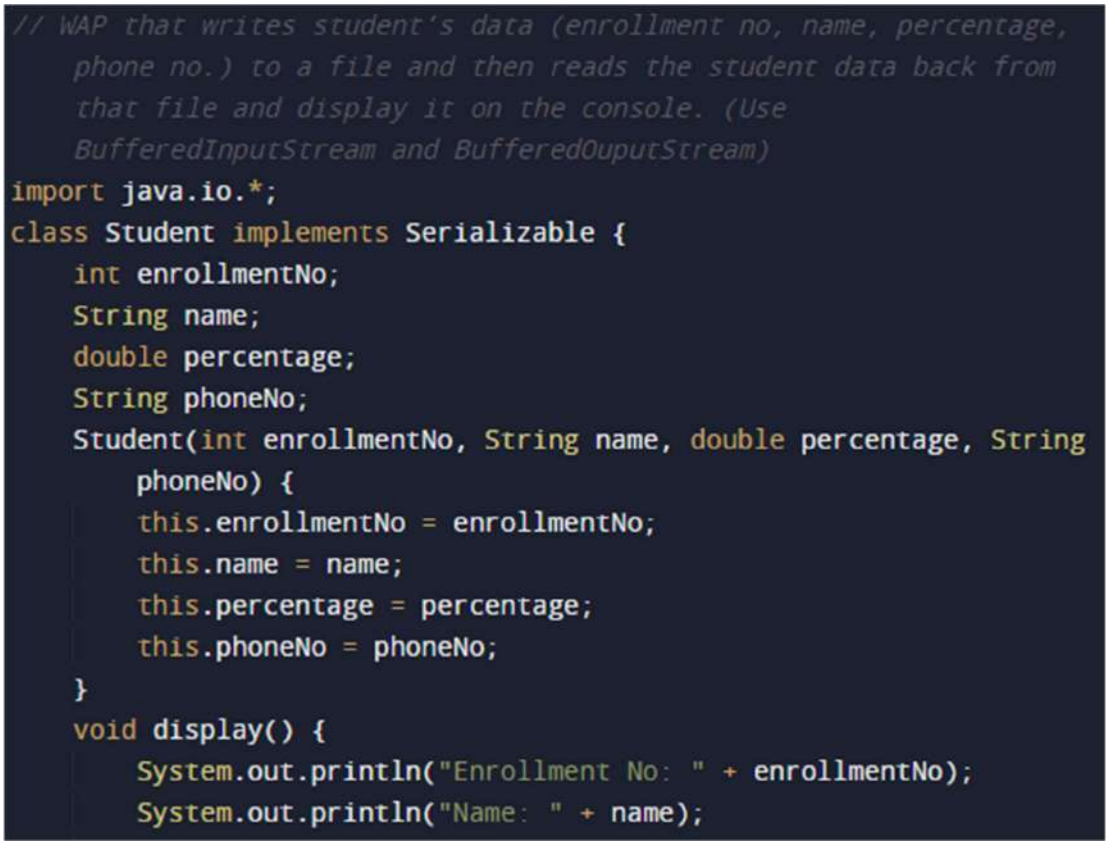
```
Done writing
```

OUTPUT:

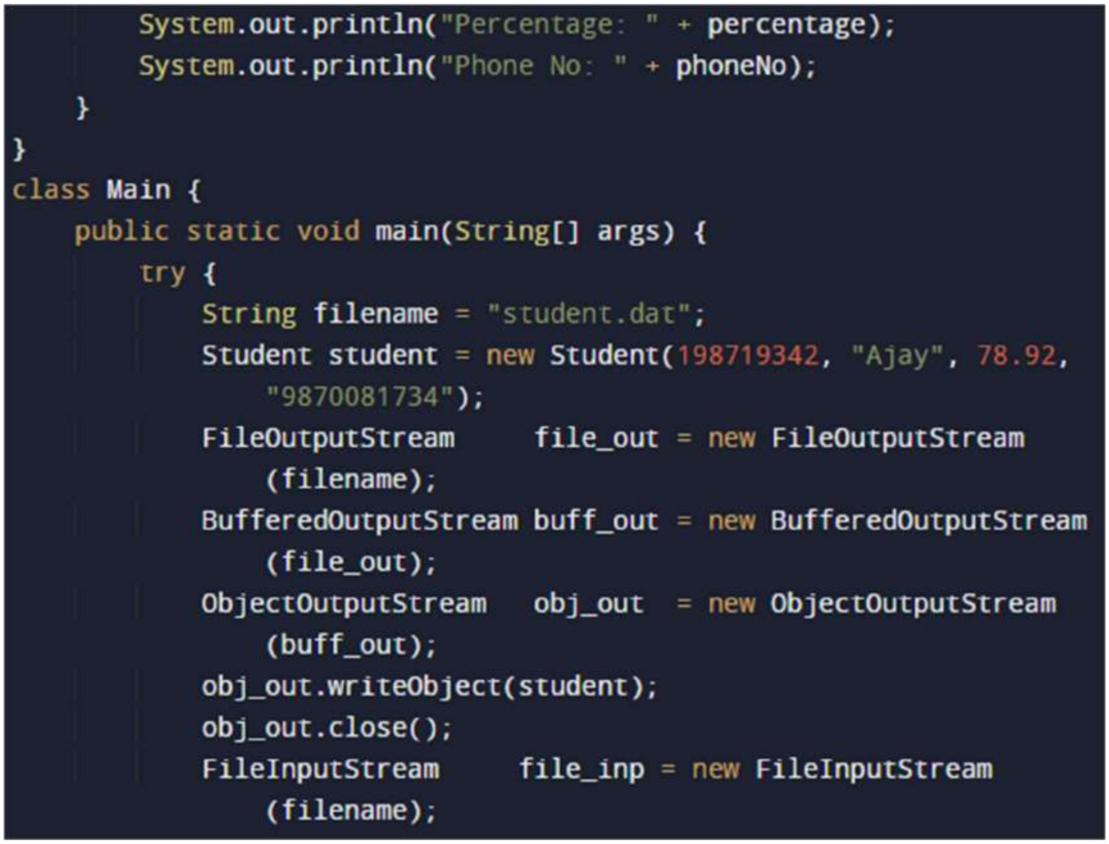


```
10
```

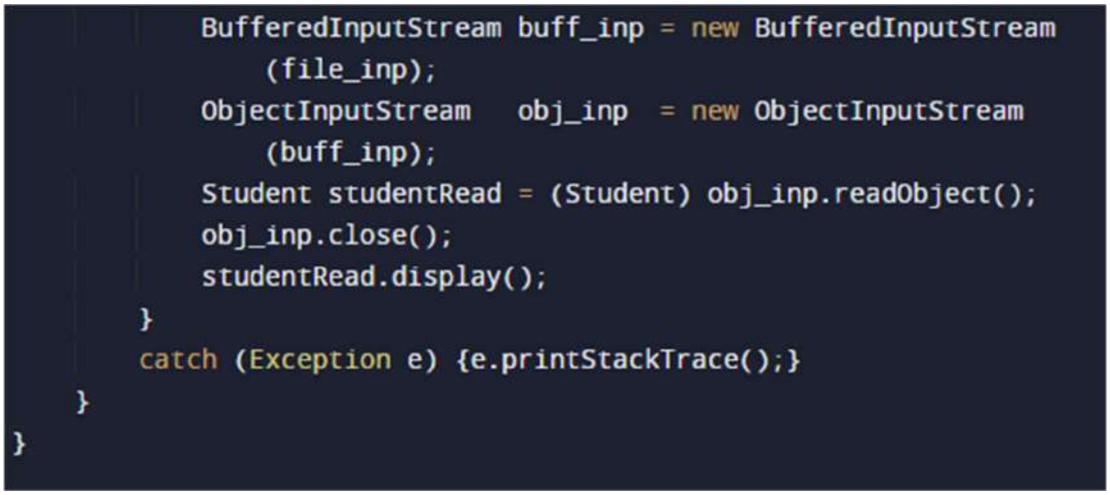
Ques24: WAP that writes student's data (enrollment no, name, percentage, phone no.) to a file and then reads the student data back from that file and display it on the console. (Use BufferedInputStream and BufferedOuputStream).



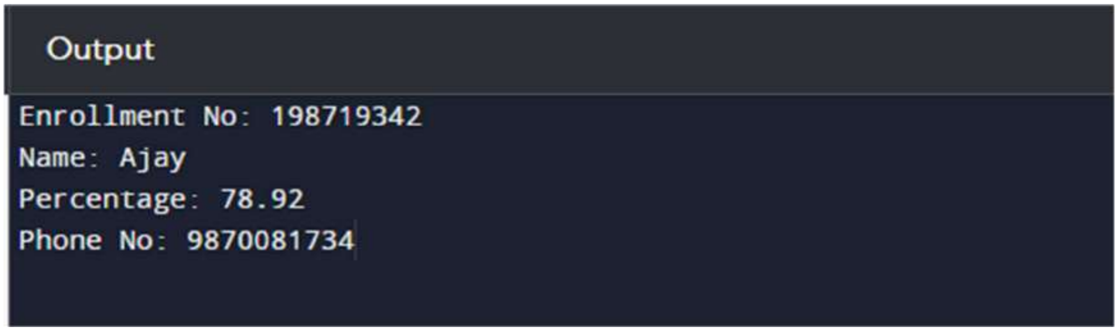
```
// WAP that writes student's data (enrollment no, name, percentage,
    phone no.) to a file and then reads the student data back from
    that file and display it on the console. (Use
    BufferedInputStream and BufferedOuputStream)
import java.io.*;
class Student implements Serializable {
    int enrollmentNo;
    String name;
    double percentage;
    String phoneNo;
    Student(int enrollmentNo, String name, double percentage, String
        phoneNo) {
        this.enrollmentNo = enrollmentNo;
        this.name = name;
        this.percentage = percentage;
        this.phoneNo = phoneNo;
    }
    void display() {
        System.out.println("Enrollment No: " + enrollmentNo);
        System.out.println("Name: " + name);
```



```
        System.out.println("Percentage: " + percentage);
        System.out.println("Phone No: " + phoneNo);
    }
}
class Main {
    public static void main(String[] args) {
        try {
            String filename = "student.dat";
            Student student = new Student(198719342, "Ajay", 78.92,
                "9870081734");
            FileOutputStream     file_out = new FileOutputStream
                (filename);
            BufferedOutputStream buff_out = new BufferedOutputStream
                (file_out);
            ObjectOutputStream   obj_out  = new ObjectOutputStream
                (buff_out);
            obj_out.writeObject(student);
            obj_out.close();
            FileInputStream     file_inp = new FileInputStream
                (filename);
```



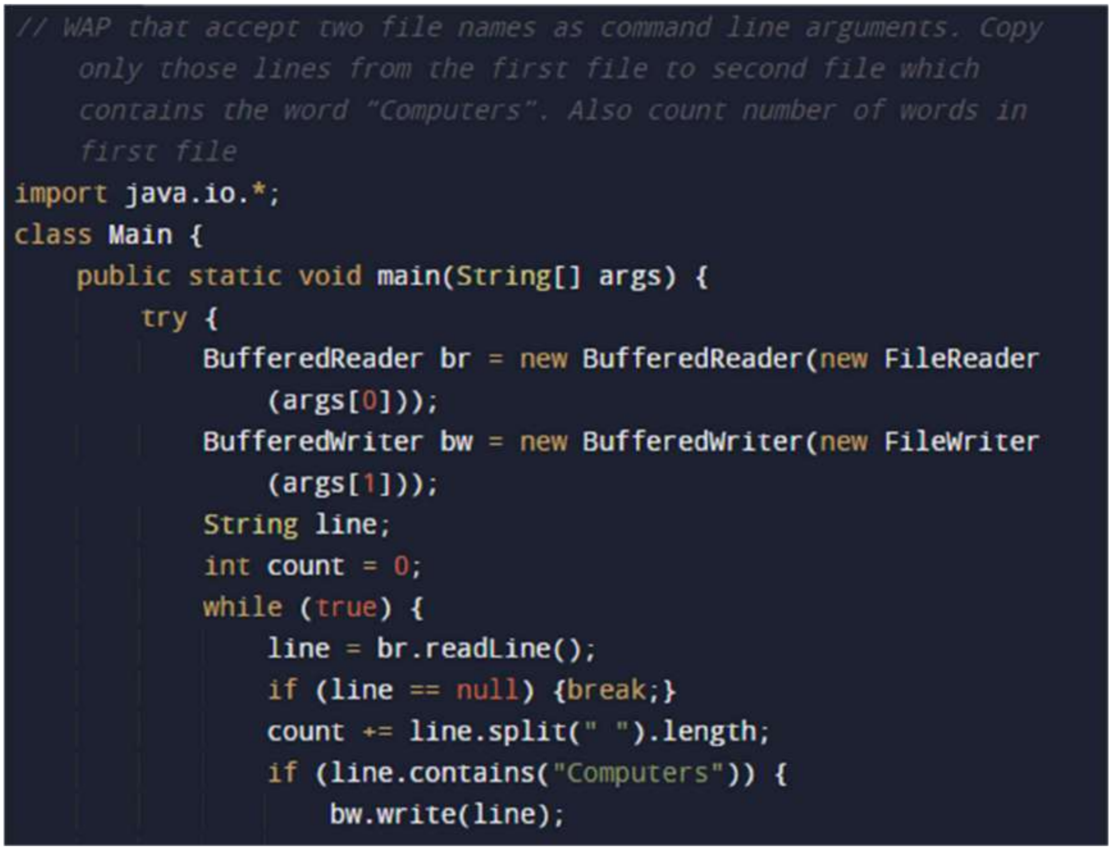
```
            BufferedInputStream buff_inp = new BufferedInputStream
                (file_inp);
            ObjectInputStream   obj_inp  = new ObjectInputStream
                (buff_inp);
            Student studentRead = (Student) obj_inp.readObject();
            obj_inp.close();
            studentRead.display();
        }
        catch (Exception e) {e.printStackTrace();}
    }
}
```



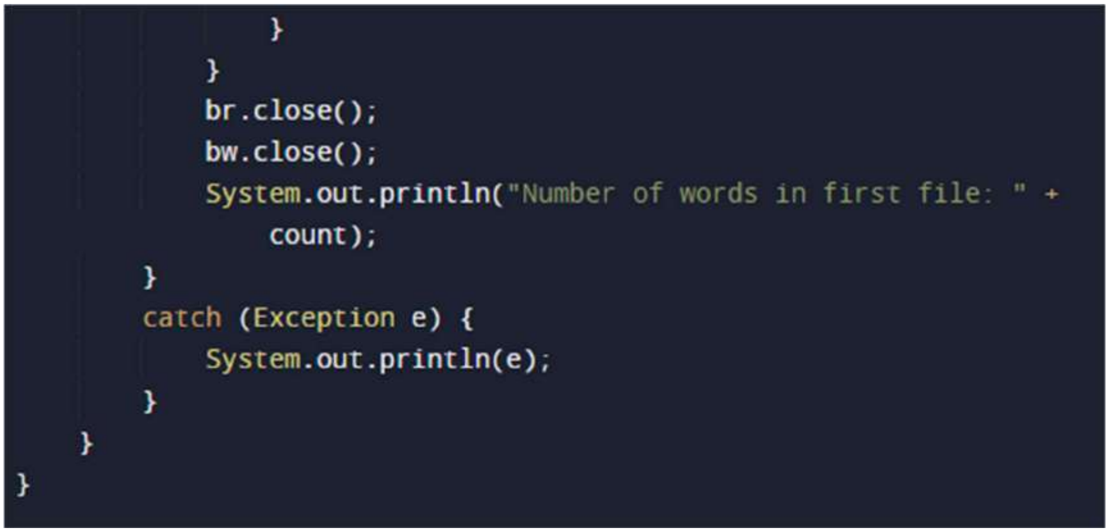
Output

```
Enrollment No: 198719342
Name: Ajay
Percentage: 78.92
Phone No: 9870081734
```

Ques25: WAP that accept two file names as command line arguments. Copy only those lines from the first file to second file which contains the word “Computers”. Also count number of words in first file.

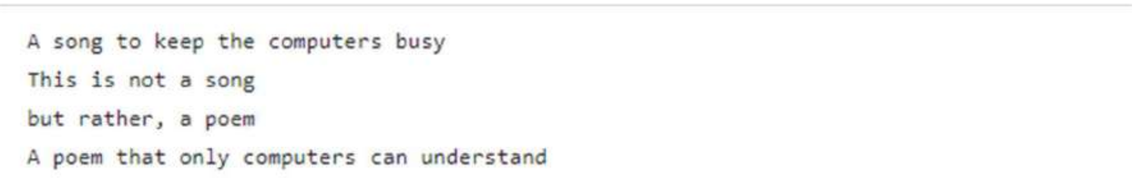


```
// WAP that accept two file names as command line arguments. Copy
    only those lines from the first file to second file which
    contains the word “Computers”. Also count number of words in
    first file
import java.io.*;
class Main {
    public static void main(String[] args) {
        try {
            BufferedReader br = new BufferedReader(new FileReader
                (args[0]));
            BufferedWriter bw = new BufferedWriter(new FileWriter
                (args[1]));
            String line;
            int count = 0;
            while (true) {
                line = br.readLine();
                if (line == null) {break;}
                count += line.split(" ").length;
                if (line.contains("Computers")) {
                    bw.write(line);
```

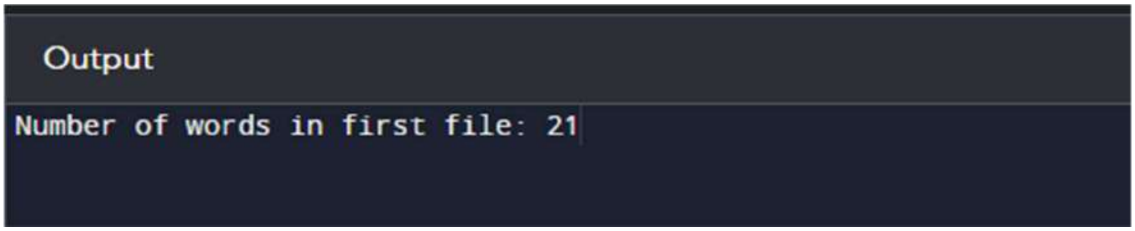


```
                }
            }
            br.close();
            bw.close();
            System.out.println("Number of words in first file: " +
                count);
        }
        catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

INPUT:

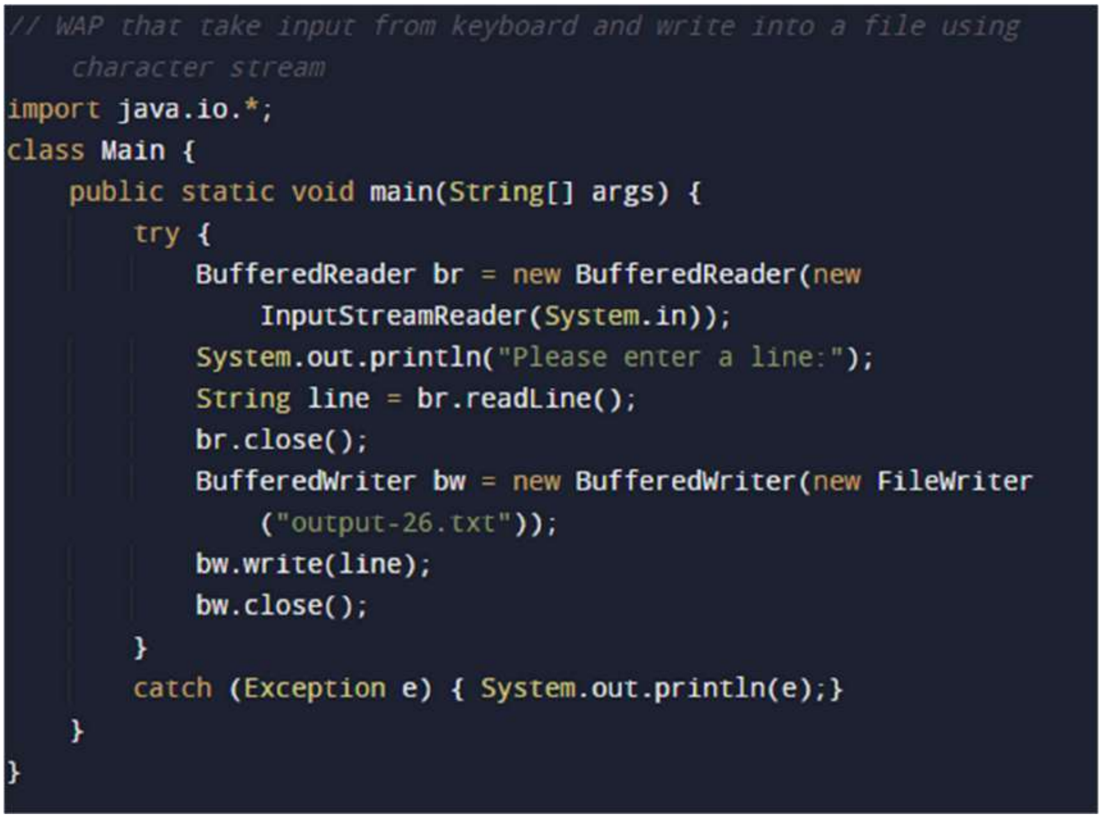


```
A song to keep the computers busy
This is not a song
but rather, a poem
A poem that only computers can understand
```



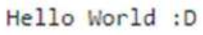
```
Number of words in first file: 21
```

Ques26: WAP that take input from keyboard and write into a file using character stream.



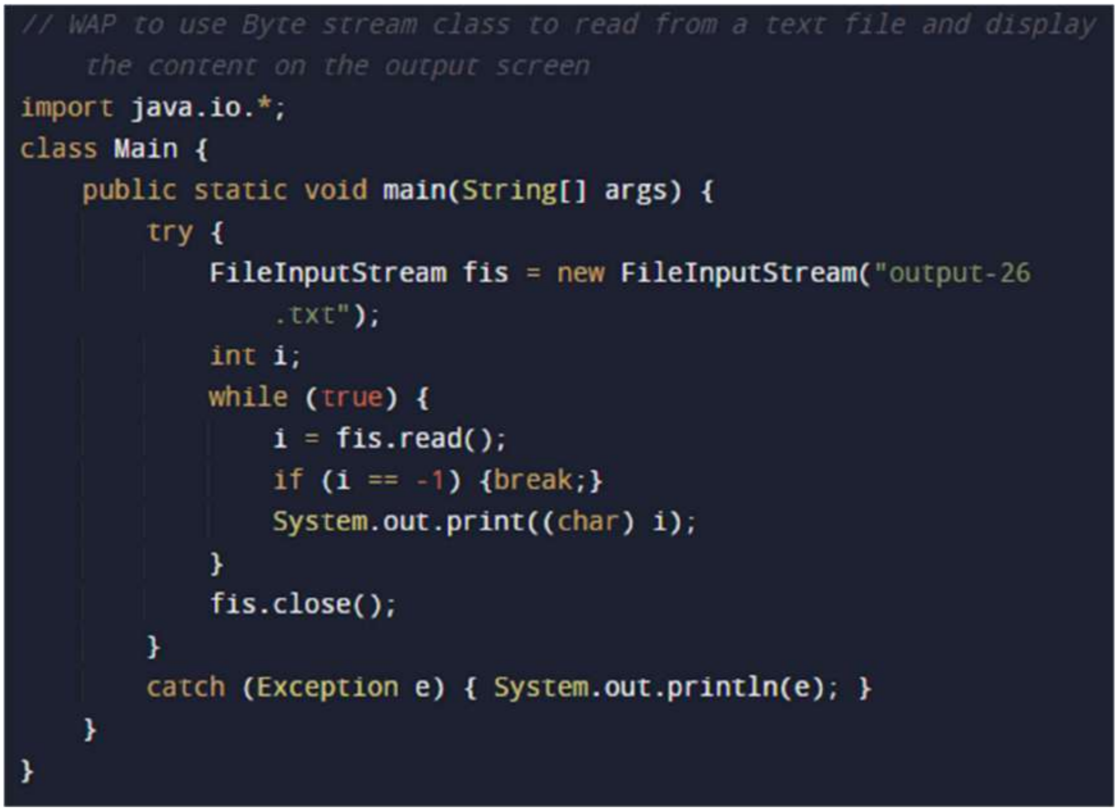
```
// WAP that take input from keyboard and write into a file using
    character stream
import java.io.*;
class Main {
    public static void main(String[] args) {
        try {
            BufferedReader br = new BufferedReader(new
                InputStreamReader(System.in));
            System.out.println("Please enter a line:");
            String line = br.readLine();
            br.close();
            BufferedWriter bw = new BufferedWriter(new FileWriter
                ("output-26.txt"));
            bw.write(line);
            bw.close();
        }
        catch (Exception e) { System.out.println(e);}
    }
}
```

OUTPUT:

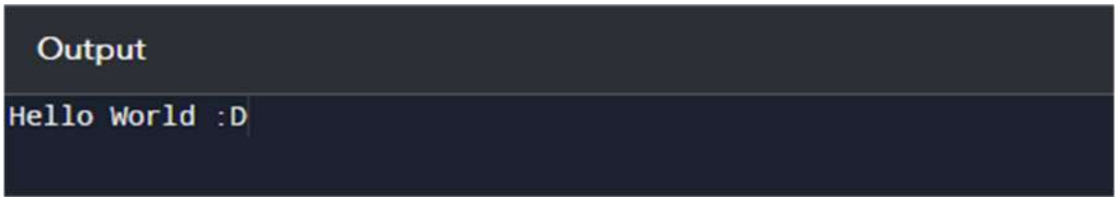


```
Hello World :D
```

Ques27: WAP to use Byte stream class to read from a text file and display the content on the output screen.

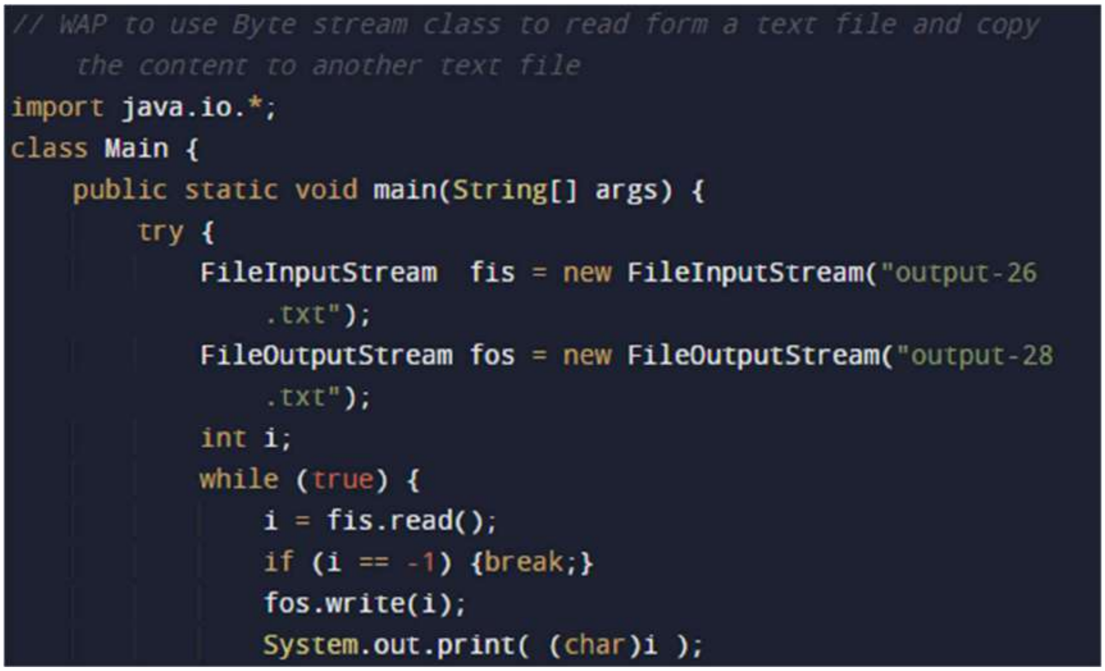


```java
// WAP to use Byte stream class to read from a text file and display
    the content on the output screen
import java.io.*;
class Main {
    public static void main(String[] args) {
        try {
            FileInputStream fis = new FileInputStream("output-26
                .txt");
            int i;
            while (true) {
                i = fis.read();
                if (i == -1) {break;}
                System.out.print((char) i);
            }
            fis.close();
        }
        catch (Exception e) { System.out.println(e); }
    }
}
```

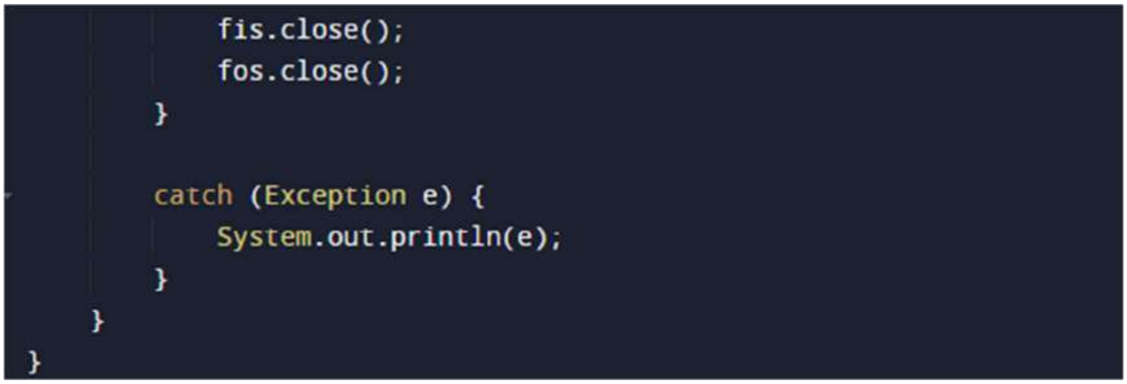


```
Hello World :D
```

Ques28: WAP to use Byte stream class to read form a text file and copy the content to another text file.

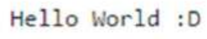


```
// WAP to use Byte stream class to read form a text file and copy
    the content to another text file
import java.io.*;
class Main {
    public static void main(String[] args) {
        try {
            FileInputStream  fis = new FileInputStream("output-26
                .txt");
            FileOutputStream fos = new FileOutputStream("output-28
                .txt");
            int i;
            while (true) {
                i = fis.read();
                if (i == -1) {break;}
                fos.write(i);
                System.out.print( (char)i );
```



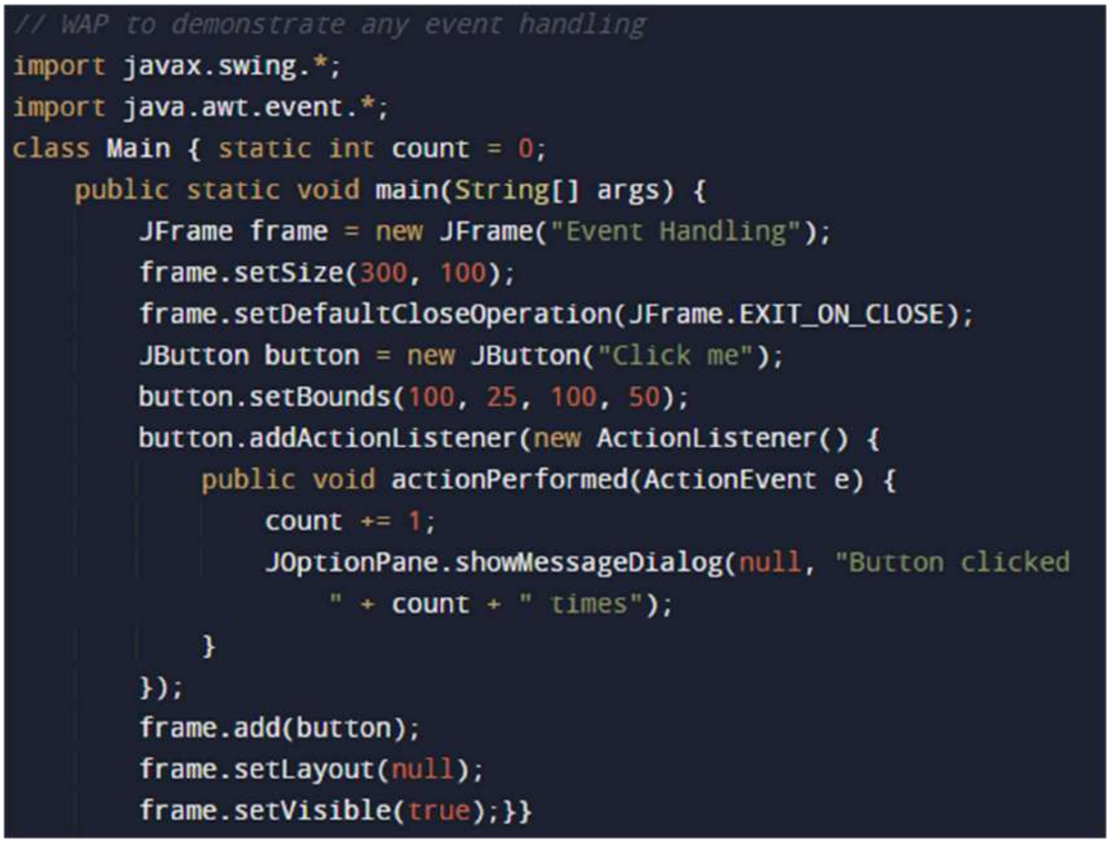
```
            fis.close();
            fos.close();
        }

        catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

OUTPUT:

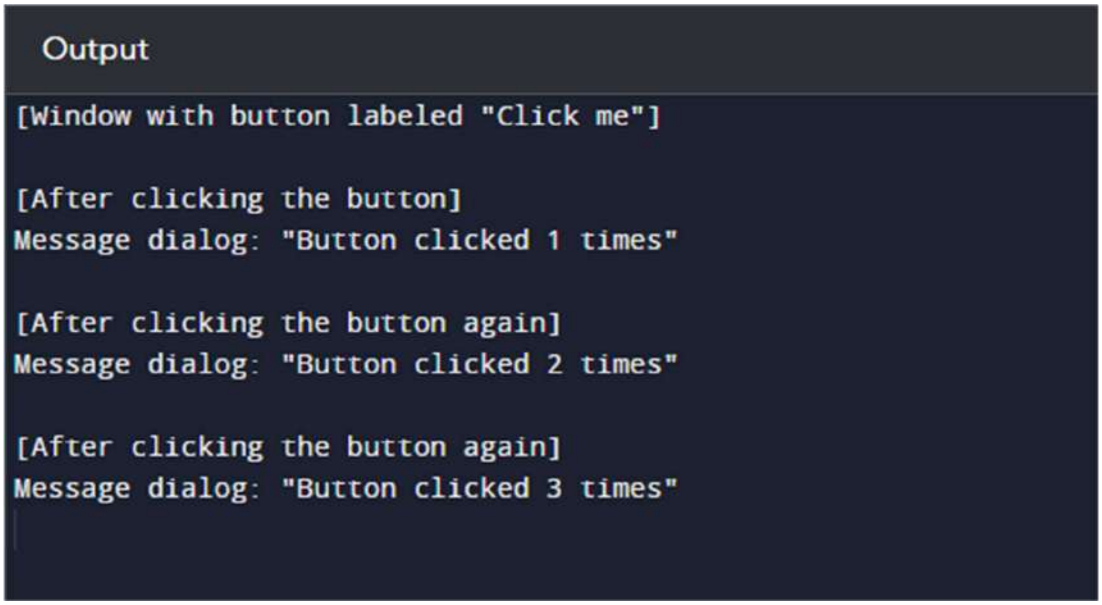


```
Hello World :D
```

Ques29: WAP to demonstrate any event handling.



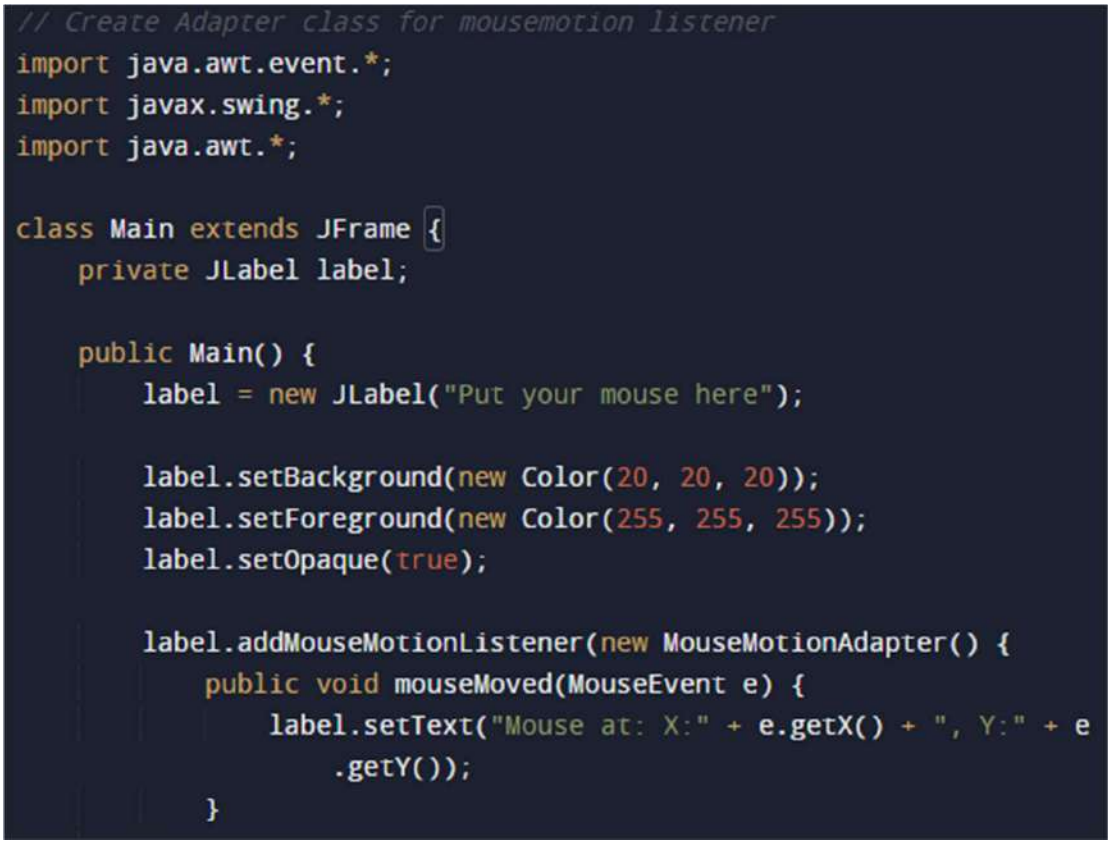
```java
// WAP to demonstrate any event handling
import javax.swing.*;
import java.awt.event.*;
class Main { static int count = 0;
    public static void main(String[] args) {
        JFrame frame = new JFrame("Event Handling");
        frame.setSize(300, 100);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JButton button = new JButton("Click me");
        button.setBounds(100, 25, 100, 50);
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                count += 1;
                JOptionPane.showMessageDialog(null, "Button clicked
                    " + count + " times");
            }
        });
        frame.add(button);
        frame.setLayout(null);
        frame.setVisible(true);}}
```



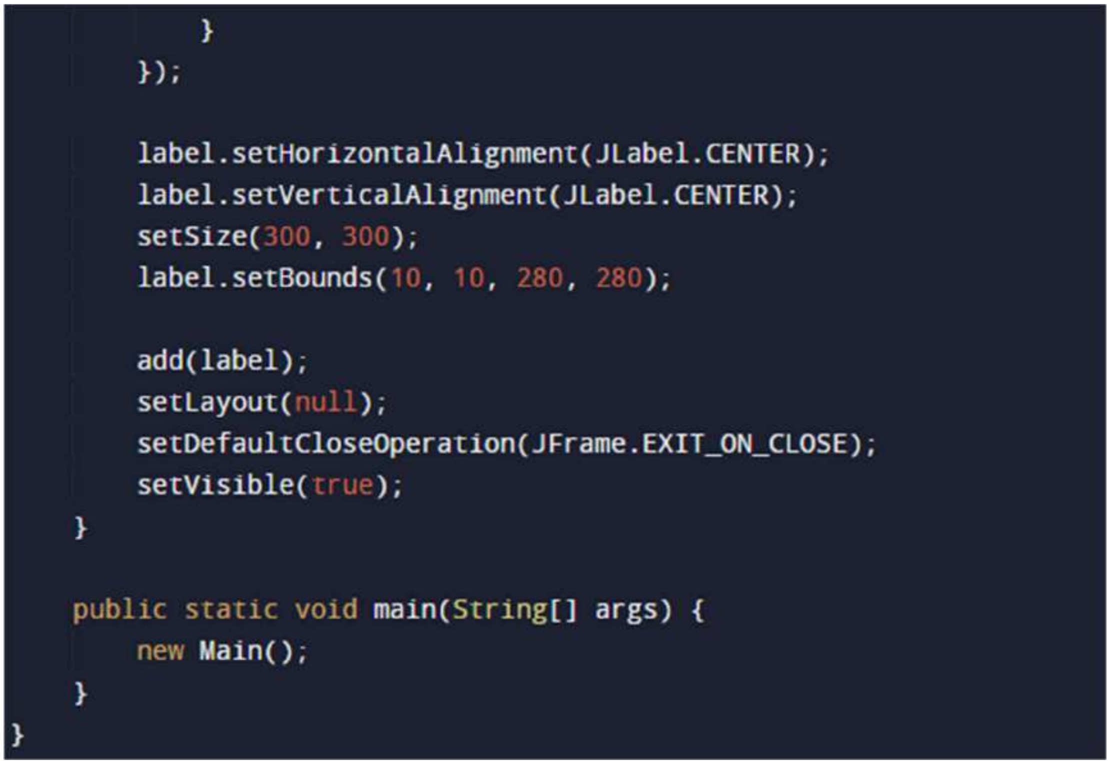
Output

```
[Window with button labeled "Click me"]

[After clicking the button]
Message dialog: "Button clicked 1 times"

[After clicking the button again]
Message dialog: "Button clicked 2 times"

[After clicking the button again]
Message dialog: "Button clicked 3 times"
```

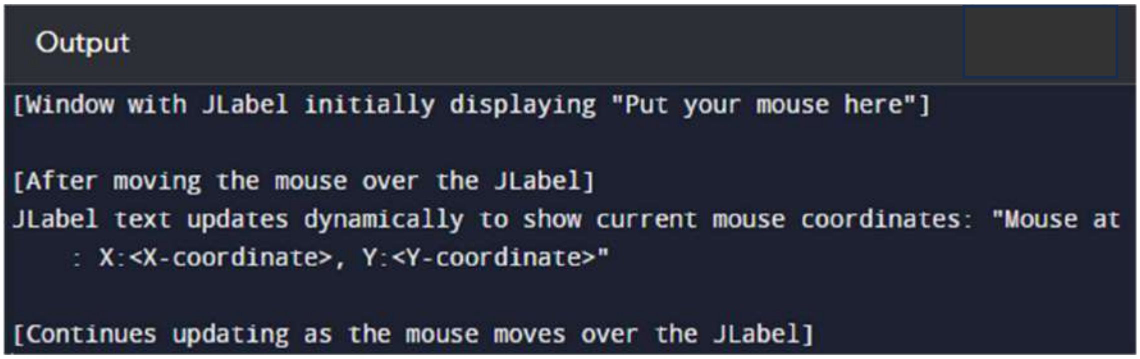
Ques30: Create Adapter class for mousemotion listener.



```
// Create Adapter class for mousemotion listener
import java.awt.event.*;
import javax.swing.*;
import java.awt.*;

class Main extends JFrame {
    private JLabel label;

    public Main() {
        label = new JLabel("Put your mouse here");

        label.setBackground(new Color(20, 20, 20));
        label.setForeground(new Color(255, 255, 255));
        label.setOpaque(true);

        label.addMouseMotionListener(new MouseMotionAdapter() {
            public void mouseMoved(MouseEvent e) {
                label.setText("Mouse at: X:" + e.getX() + ", Y:" + e
                    .getY());
            }
```

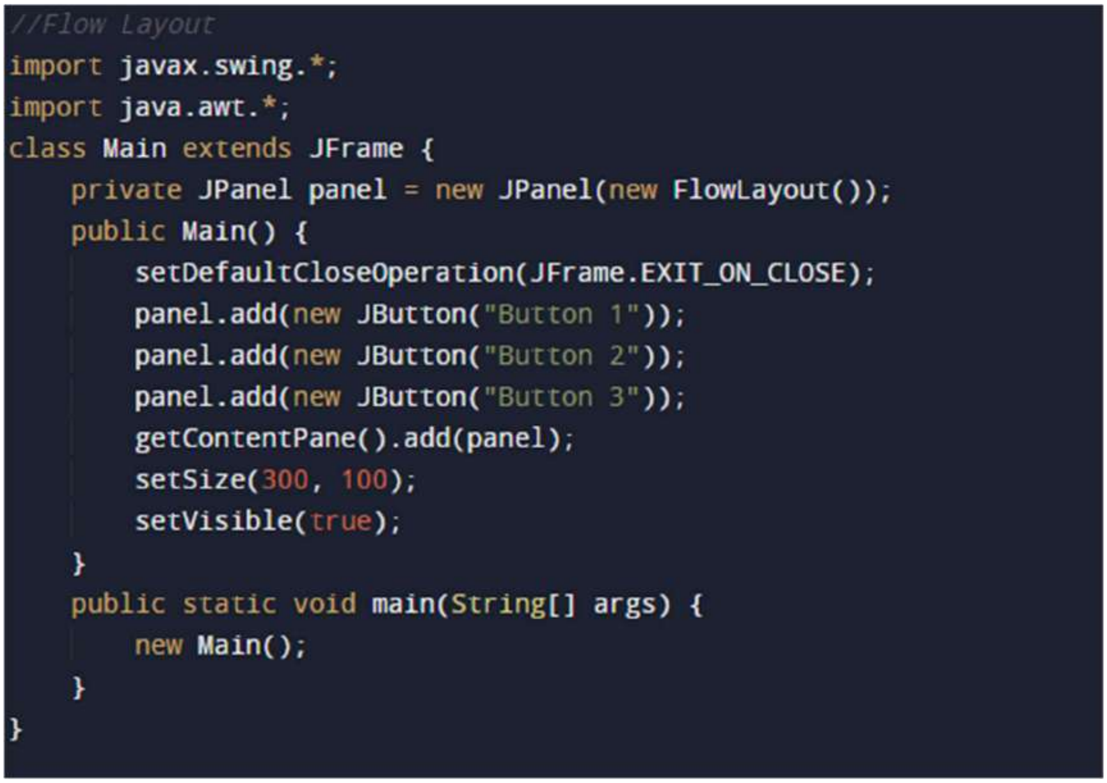


```
            }
        });

        label.setHorizontalAlignment(JLabel.CENTER);
        label.setVerticalAlignment(JLabel.CENTER);
        setSize(300, 300);
        label.setBounds(10, 10, 280, 280);

        add(label);
        setLayout(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }

    public static void main(String[] args) {
        new Main();
    }
}
```

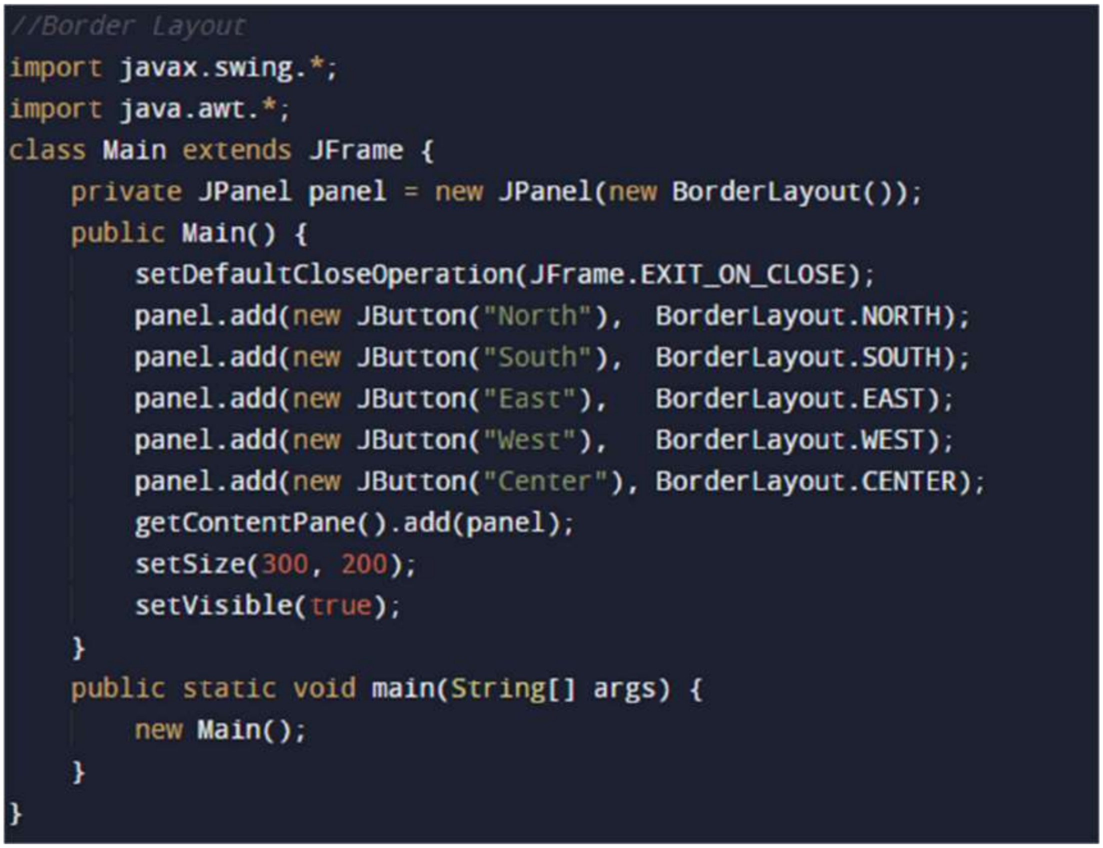


```
[Window with JLabel initially displaying "Put your mouse here"]

[After moving the mouse over the JLabel]
JLabel text updates dynamically to show current mouse coordinates: "Mouse at
    : X:<X-coordinate>, Y:<Y-coordinate>"

[Continues updating as the mouse moves over the JLabel]
```

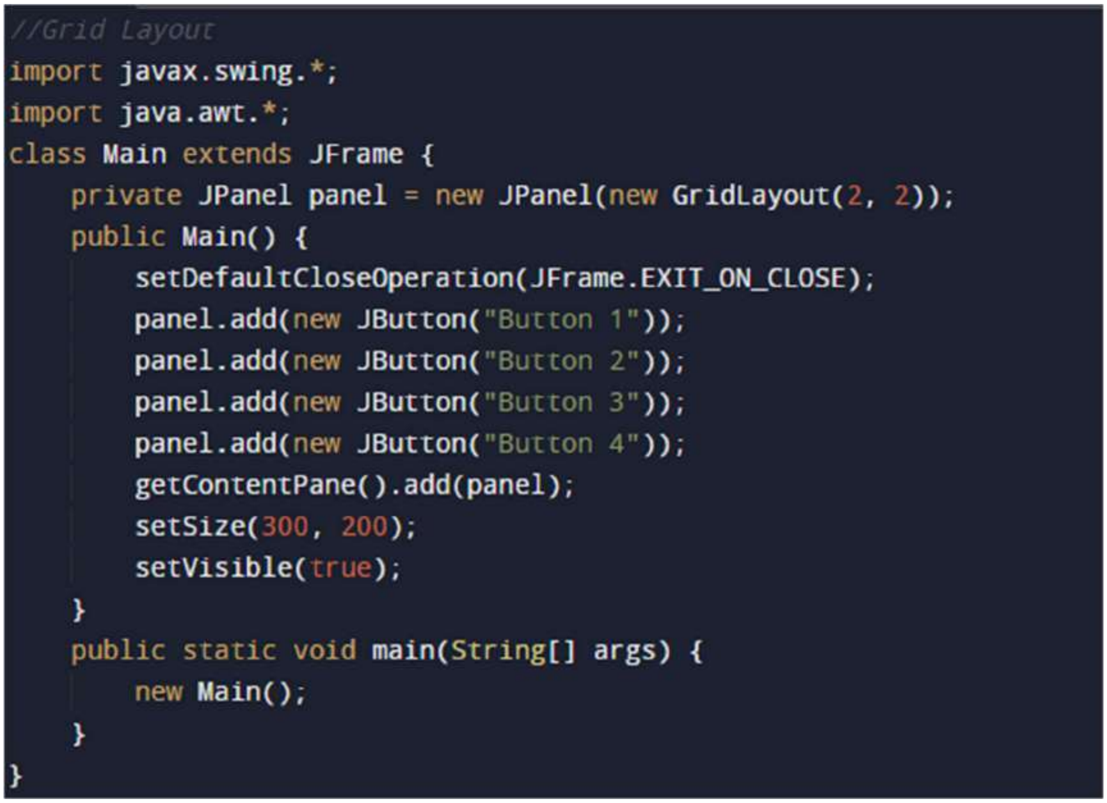
Ques31: Write 4 different programs to implement all 4 layouts (Swings).



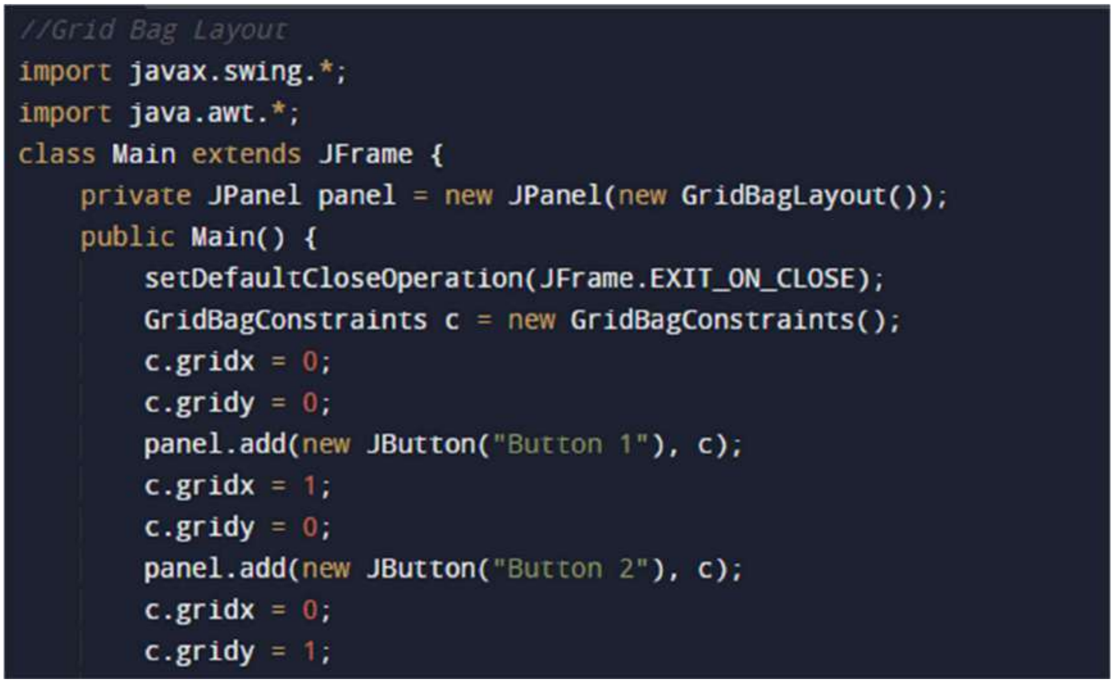
```java
//Flow Layout
import javax.swing.*;
import java.awt.*;
class Main extends JFrame {
    private JPanel panel = new JPanel(new FlowLayout());
    public Main() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        panel.add(new JButton("Button 1"));
        panel.add(new JButton("Button 2"));
        panel.add(new JButton("Button 3"));
        getContentPane().add(panel);
        setSize(300, 100);
        setVisible(true);
    }
    public static void main(String[] args) {
        new Main();
    }
}
```



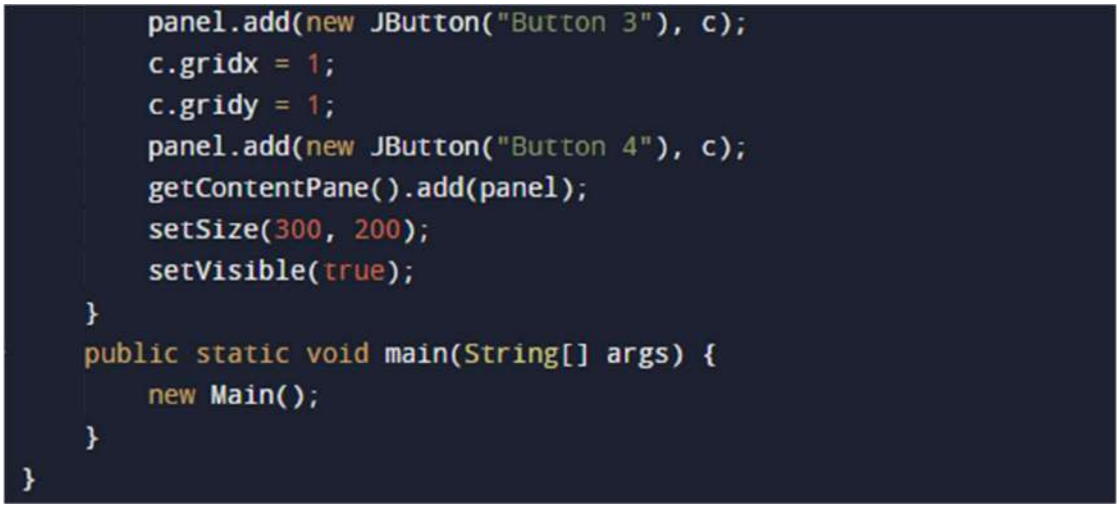
```java
//Border Layout
import javax.swing.*;
import java.awt.*;
class Main extends JFrame {
    private JPanel panel = new JPanel(new BorderLayout());
    public Main() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        panel.add(new JButton("North"),  BorderLayout.NORTH);
        panel.add(new JButton("South"),  BorderLayout.SOUTH);
        panel.add(new JButton("East"),   BorderLayout.EAST);
        panel.add(new JButton("West"),   BorderLayout.WEST);
        panel.add(new JButton("Center"), BorderLayout.CENTER);
        getContentPane().add(panel);
        setSize(300, 200);
        setVisible(true);
    }
    public static void main(String[] args) {
        new Main();
    }
}
```



```
//Grid Layout
import javax.swing.*;
import java.awt.*;
class Main extends JFrame {
    private JPanel panel = new JPanel(new GridLayout(2, 2));
    public Main() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        panel.add(new JButton("Button 1"));
        panel.add(new JButton("Button 2"));
        panel.add(new JButton("Button 3"));
        panel.add(new JButton("Button 4"));
        getContentPane().add(panel);
        setSize(300, 200);
        setVisible(true);
    }
    public static void main(String[] args) {
        new Main();
    }
}
```



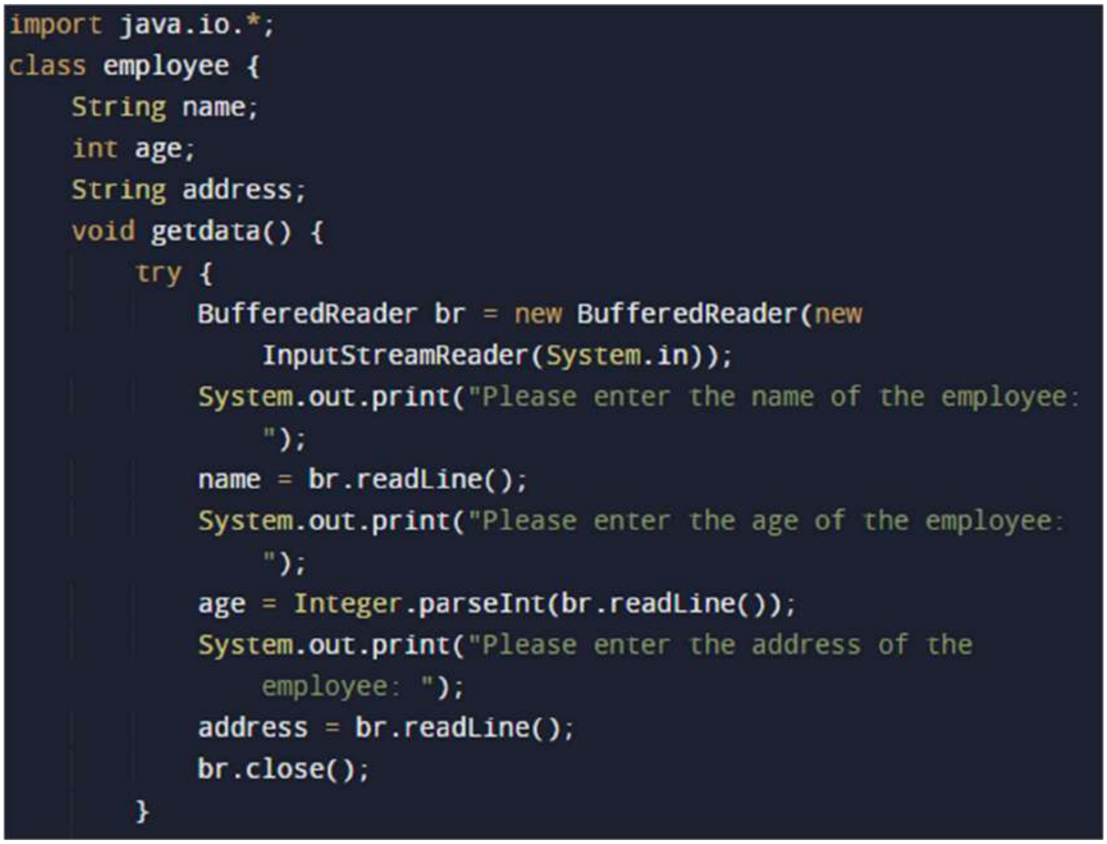
```
//Grid Bag Layout
import javax.swing.*;
import java.awt.*;
class Main extends JFrame {
    private JPanel panel = new JPanel(new GridBagLayout());
    public Main() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        GridBagConstraints c = new GridBagConstraints();
        c.gridx = 0;
        c.gridy = 0;
        panel.add(new JButton("Button 1"), c);
        c.gridx = 1;
        c.gridy = 0;
        panel.add(new JButton("Button 2"), c);
        c.gridx = 0;
        c.gridy = 1;
```



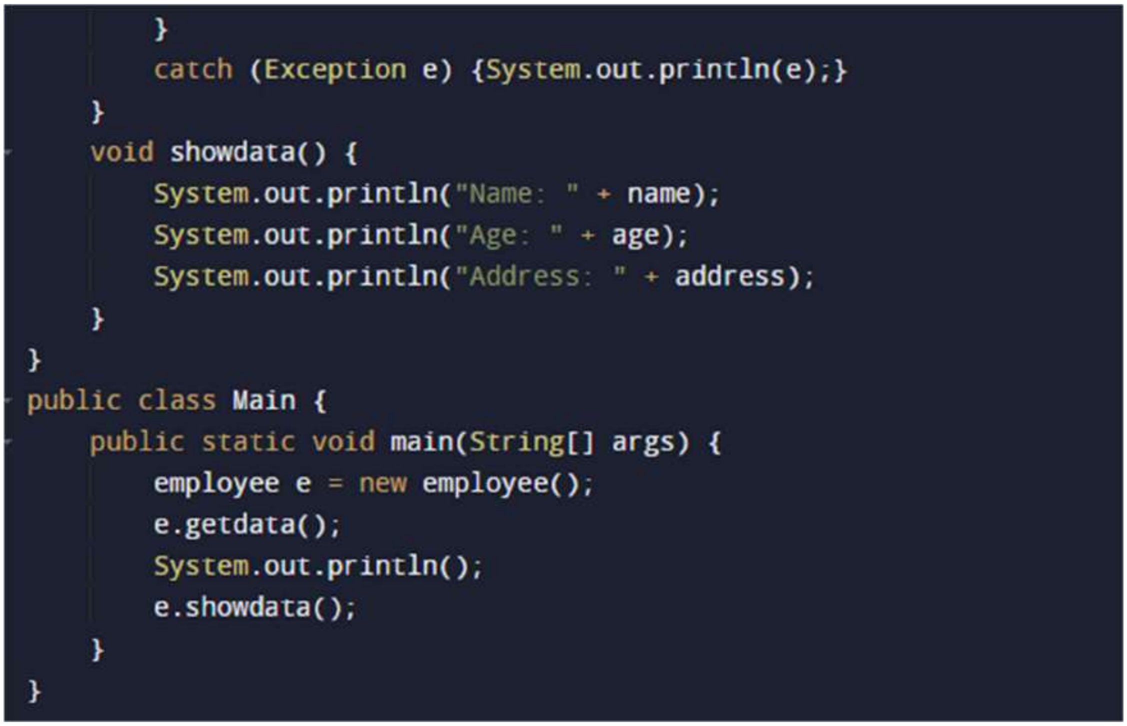
```
        panel.add(new JButton("Button 3"), c);
        c.gridx = 1;
        c.gridy = 1;
        panel.add(new JButton("Button 4"), c);
        getContentPane().add(panel);
        setSize(300, 200);
        setVisible(true);
    }
    public static void main(String[] args) {
        new Main();
    }
}
```

Ques32: Create a class employee which have name, age and address of employee, include methods getdata() and showdata(), getdata() takes the input from the user, showdata() display the data in following format:

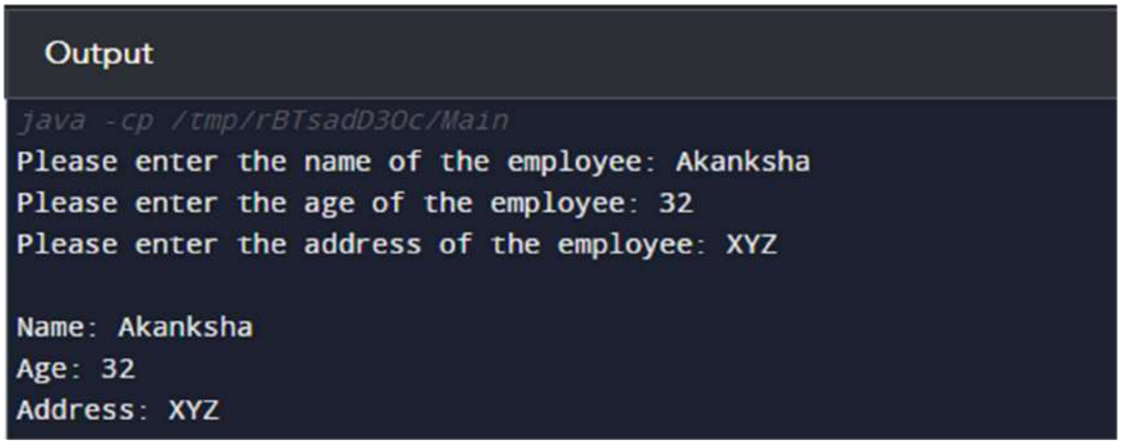
Name:
Age:
Address:



```
import java.io.*;
class employee {
    String name;
    int age;
    String address;
    void getdata() {
        try {
            BufferedReader br = new BufferedReader(new
                InputStreamReader(System.in));
            System.out.print("Please enter the name of the employee:
                ");
            name = br.readLine();
            System.out.print("Please enter the age of the employee:
                ");
            age = Integer.parseInt(br.readLine());
            System.out.print("Please enter the address of the
                employee: ");
            address = br.readLine();
            br.close();
        }
```

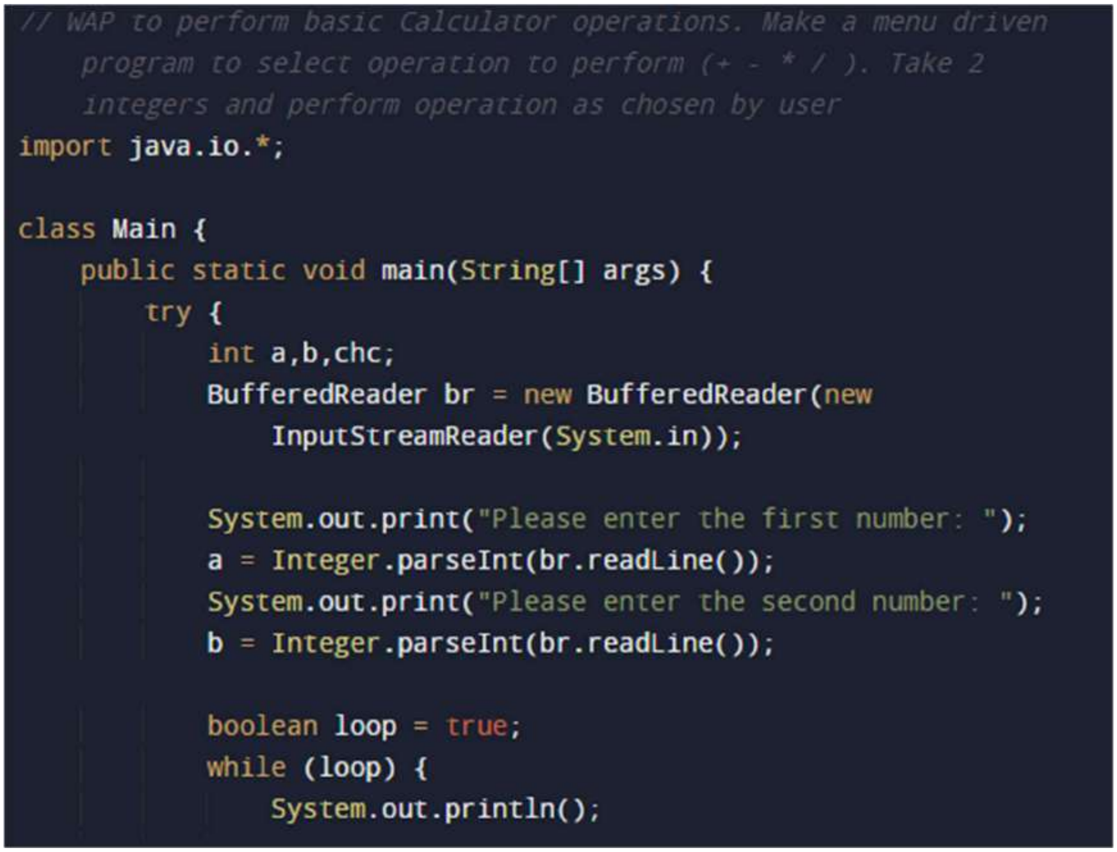


```
        }
        catch (Exception e) {System.out.println(e);}
    }
    void showdata() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Address: " + address);
    }
}
public class Main {
    public static void main(String[] args) {
        employee e = new employee();
        e.getdata();
        System.out.println();
        e.showdata();
    }
}
```

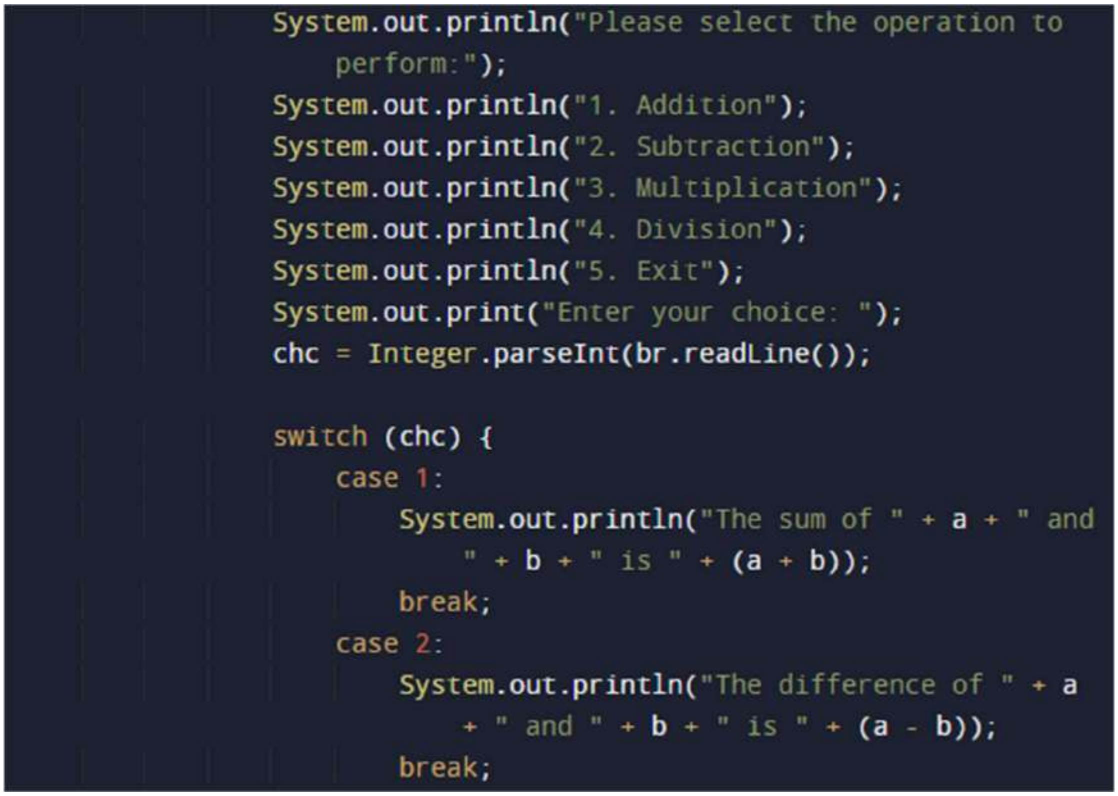


```
java -cp /tmp/rBTsadD3Oc/Main
Please enter the name of the employee: Akanksha
Please enter the age of the employee: 32
Please enter the address of the employee: XYZ

Name: Akanksha
Age: 32
Address: XYZ
```

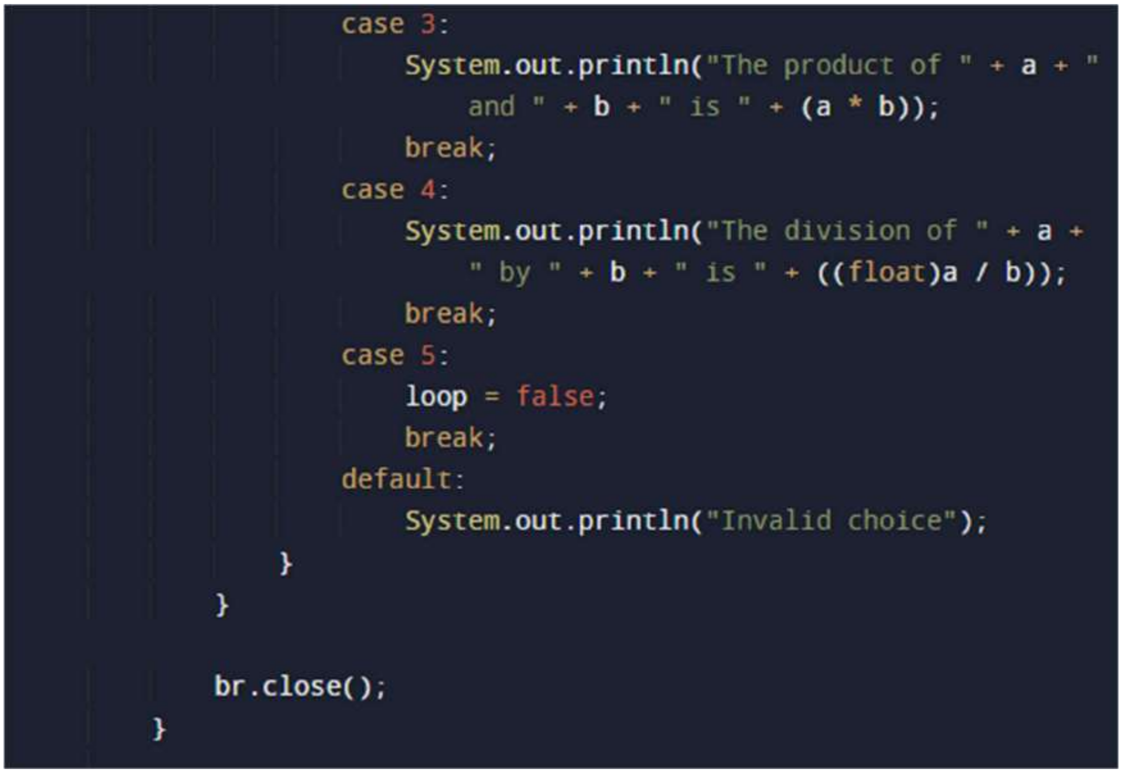
Ques33: WAP to perform basic Calculator operations. Make a menu driven program to select operation to perform (+ - * /). Take 2 integers and perform operation as chosen by user.



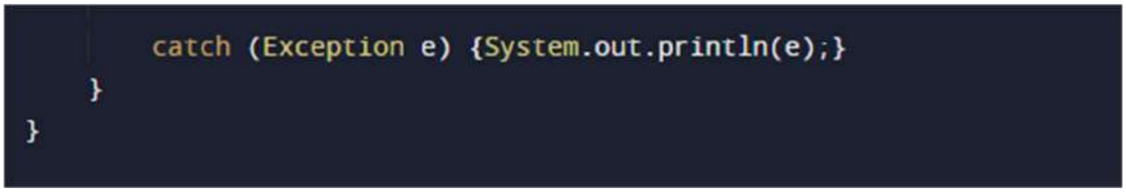
```java
// WAP to perform basic Calculator operations. Make a menu driven
    program to select operation to perform (+ - * / ). Take 2
    integers and perform operation as chosen by user
import java.io.*;

class Main {
    public static void main(String[] args) {
        try {
            int a,b,chc;
            BufferedReader br = new BufferedReader(new
                InputStreamReader(System.in));

            System.out.print("Please enter the first number: ");
            a = Integer.parseInt(br.readLine());
            System.out.print("Please enter the second number: ");
            b = Integer.parseInt(br.readLine());

            boolean loop = true;
            while (loop) {
                System.out.println();
```



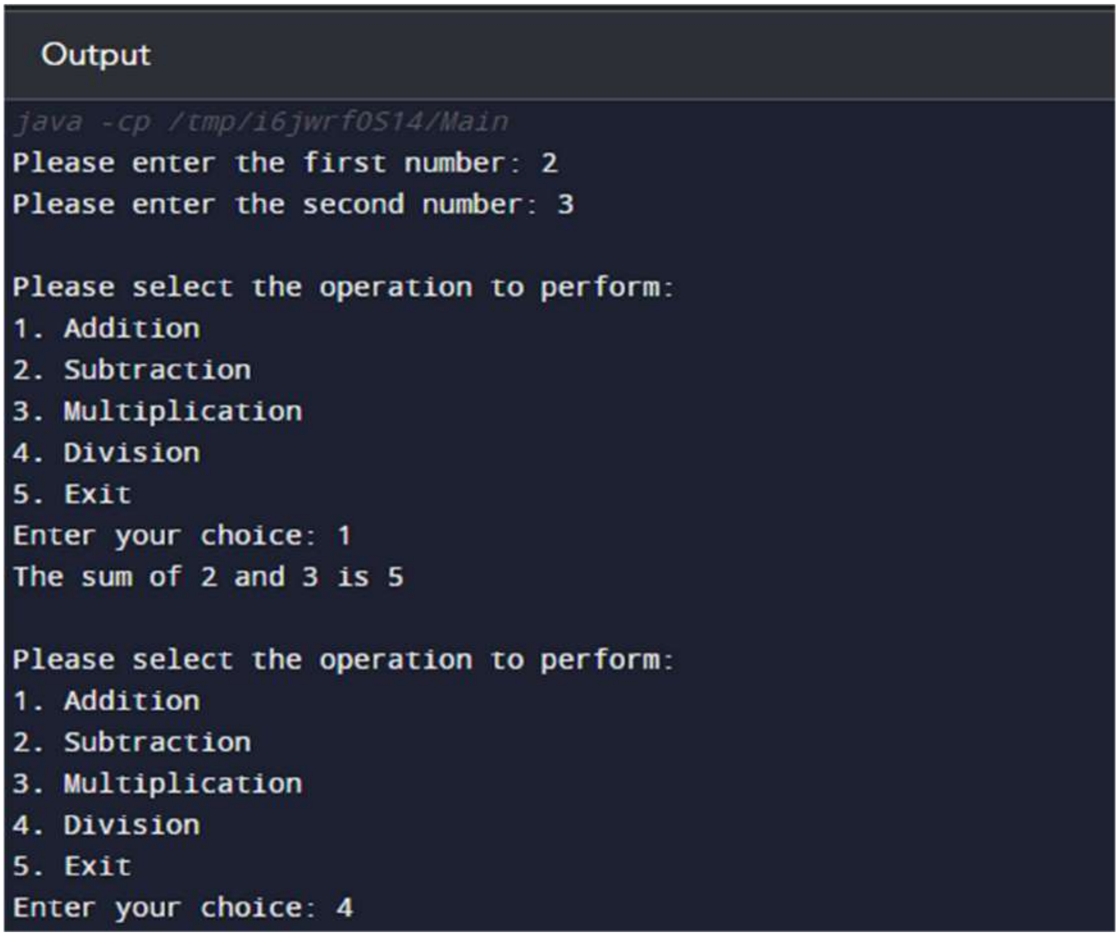
```
            System.out.println("Please select the operation to
                perform:");
            System.out.println("1. Addition");
            System.out.println("2. Subtraction");
            System.out.println("3. Multiplication");
            System.out.println("4. Division");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            chc = Integer.parseInt(br.readLine());

            switch (chc) {
                case 1:
                    System.out.println("The sum of " + a + " and
                        " + b + " is " + (a + b));
                    break;
                case 2:
                    System.out.println("The difference of " + a
                        + " and " + b + " is " + (a - b));
                    break;
```



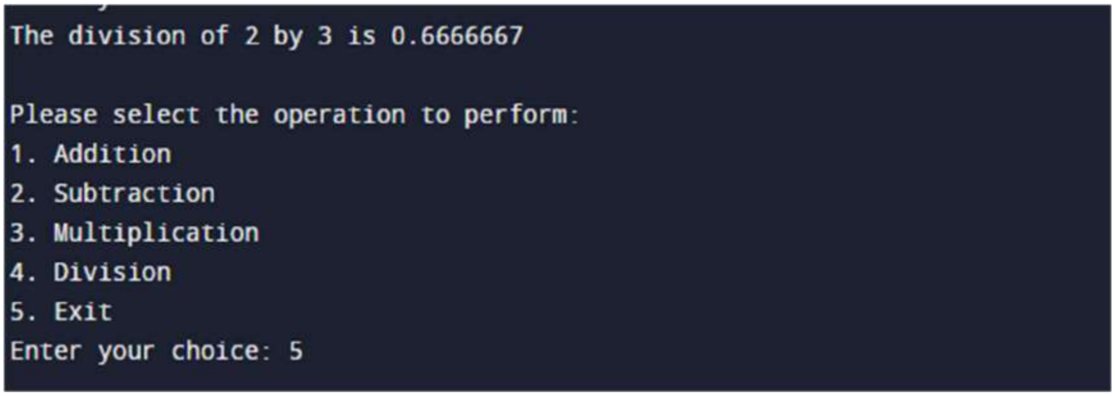
```
                case 3:
                    System.out.println("The product of " + a + "
                        and " + b + " is " + (a * b));
                    break;
                case 4:
                    System.out.println("The division of " + a +
                        " by " + b + " is " + ((float)a / b));
                    break;
                case 5:
                    loop = false;
                    break;
                default:
                    System.out.println("Invalid choice");
            }
        }

        br.close();
    }
```



```
        catch (Exception e) {System.out.println(e);}
    }
}
```

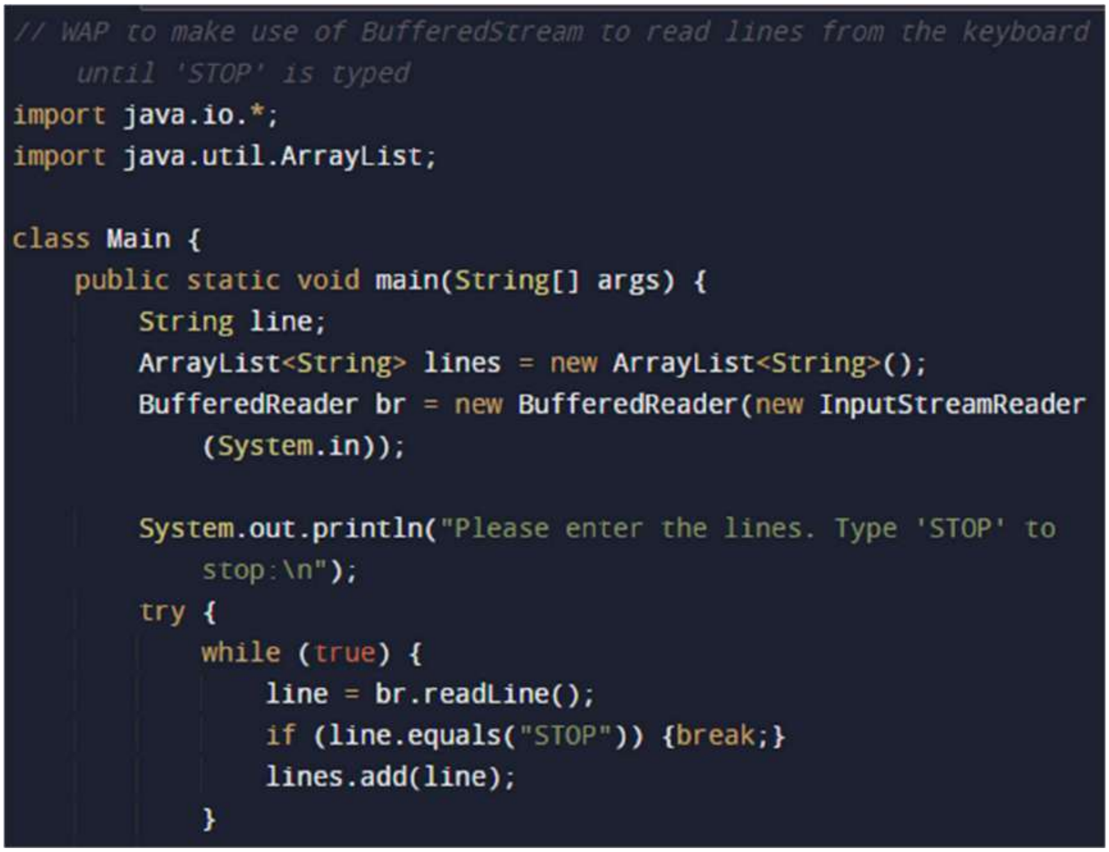


```
java -cp /tmp/i6jwrf0S14/Main
Please enter the first number: 2
Please enter the second number: 3

Please select the operation to perform:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice: 1
The sum of 2 and 3 is 5

Please select the operation to perform:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice: 4
```

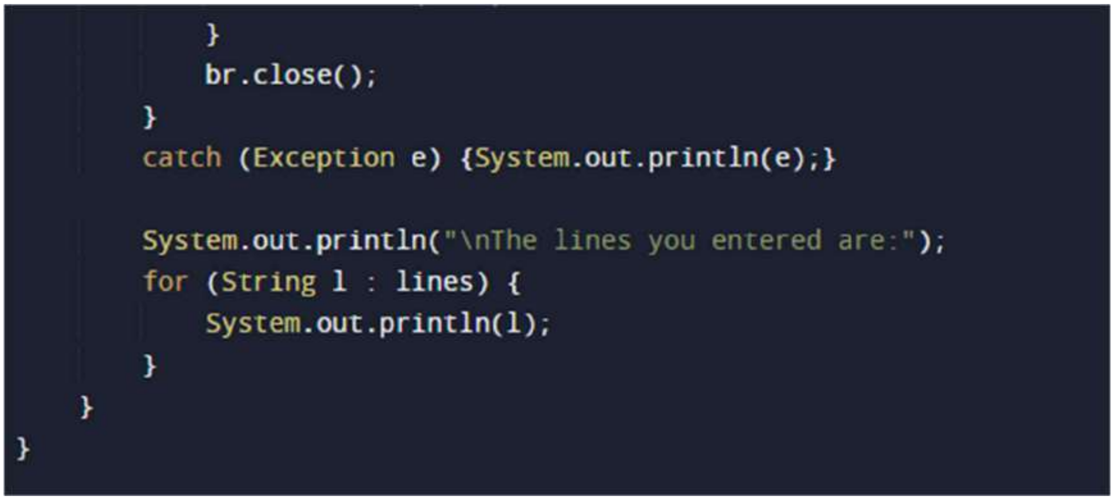


```
The division of 2 by 3 is 0.6666667

Please select the operation to perform:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice: 5
```

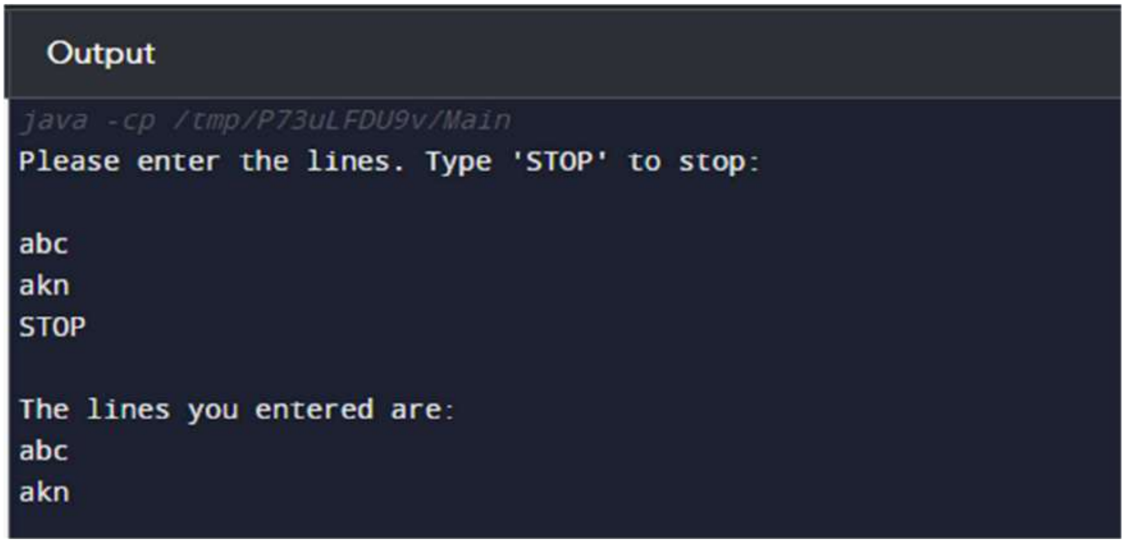
Ques34: WAP to make use of BufferedStream to read lines from the keyboard until 'STOP' is typed.



```
// WAP to make use of BufferedStream to read lines from the keyboard
    until 'STOP' is typed
import java.io.*;
import java.util.ArrayList;

class Main {
    public static void main(String[] args) {
        String line;
        ArrayList<String> lines = new ArrayList<String>();
        BufferedReader br = new BufferedReader(new InputStreamReader
            (System.in));

        System.out.println("Please enter the lines. Type 'STOP' to
            stop:\n");
        try {
            while (true) {
                line = br.readLine();
                if (line.equals("STOP")) {break;}
                lines.add(line);
            }
```

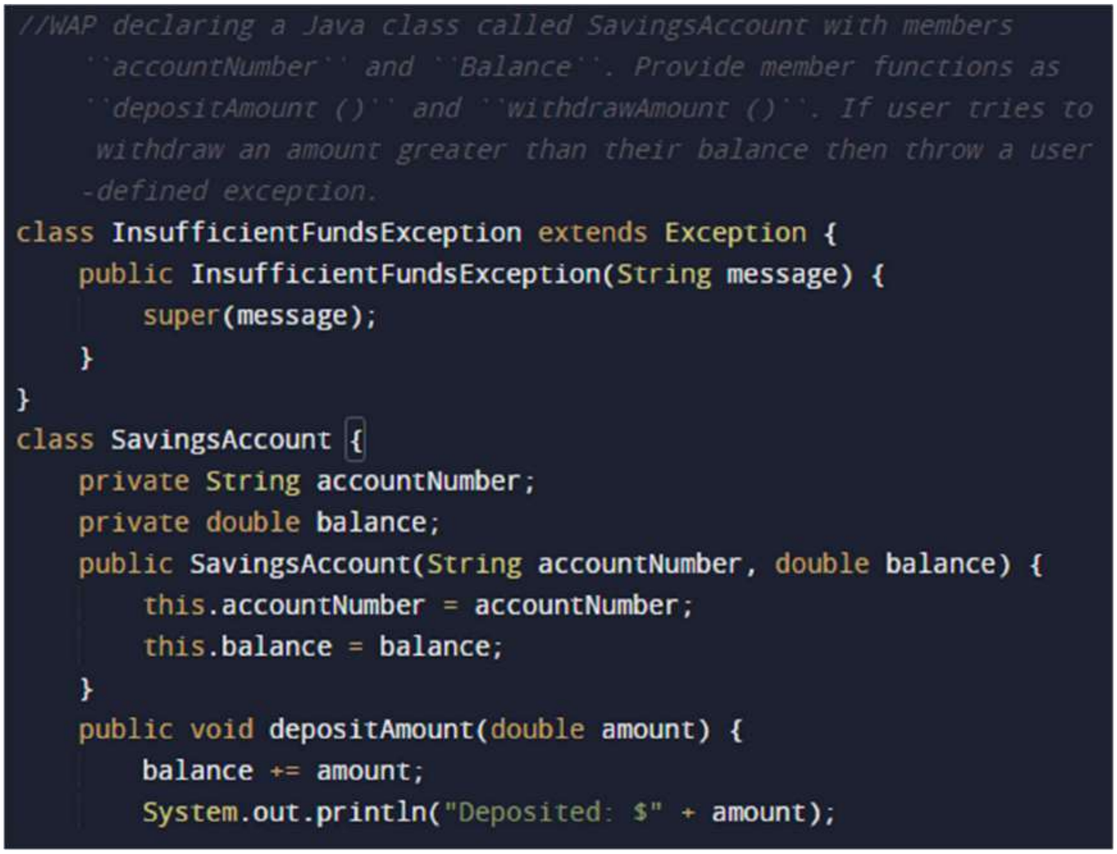


```
            }
            br.close();
        }
        catch (Exception e) {System.out.println(e);}

        System.out.println("\nThe lines you entered are:");
        for (String l : lines) {
            System.out.println(l);
        }
    }
}
```

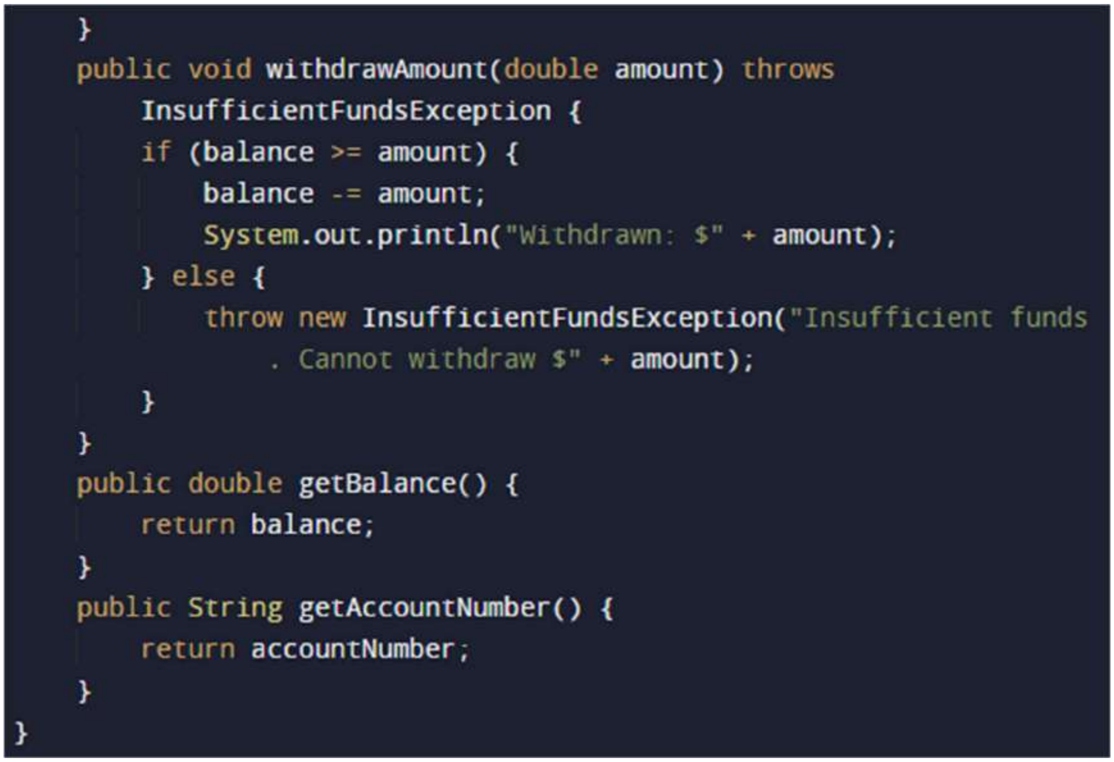


```
java -cp /tmp/P73uLFDU9v/Main
Please enter the lines. Type 'STOP' to stop:

abc
akn
STOP

The lines you entered are:
abc
akn
```

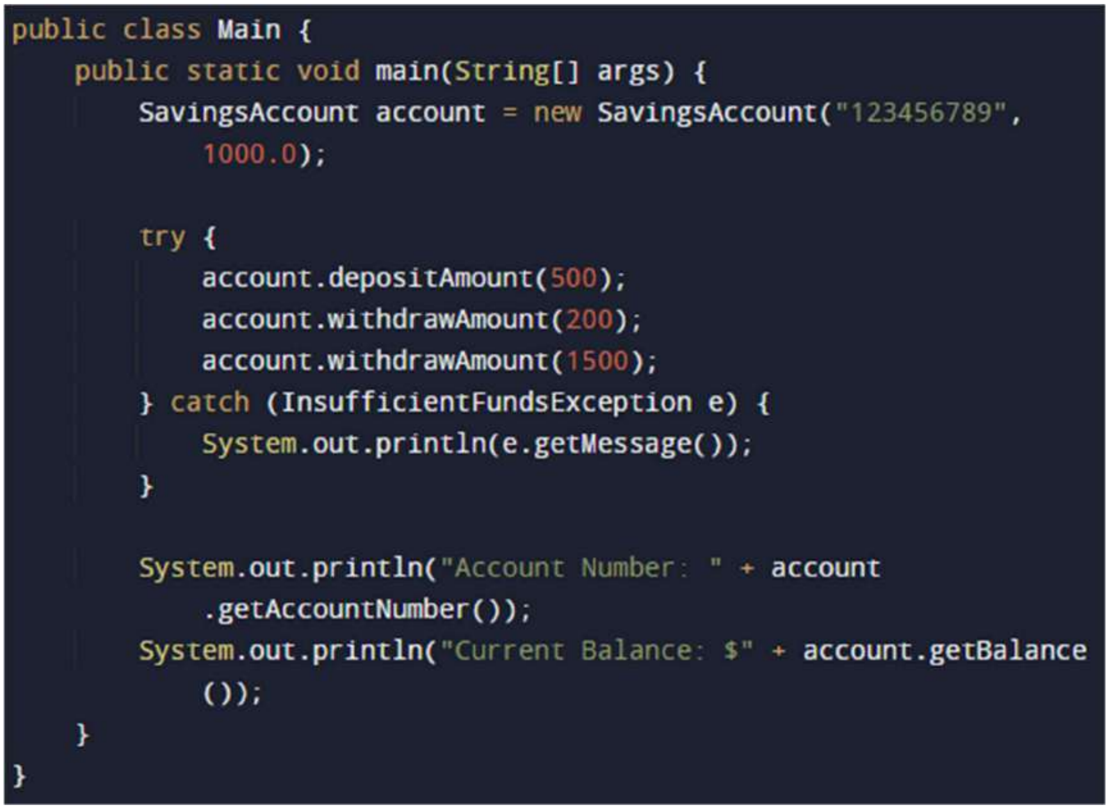
Ques35: WAP declaring a Java class called SavingsAccount with members ``accountNumber`` and ``Balance``. Provide member functions as ``depositAmount ()`` and ``withdrawAmount ()``. If user tries to withdraw an amount greater than their balance then throw a user-defined exception.



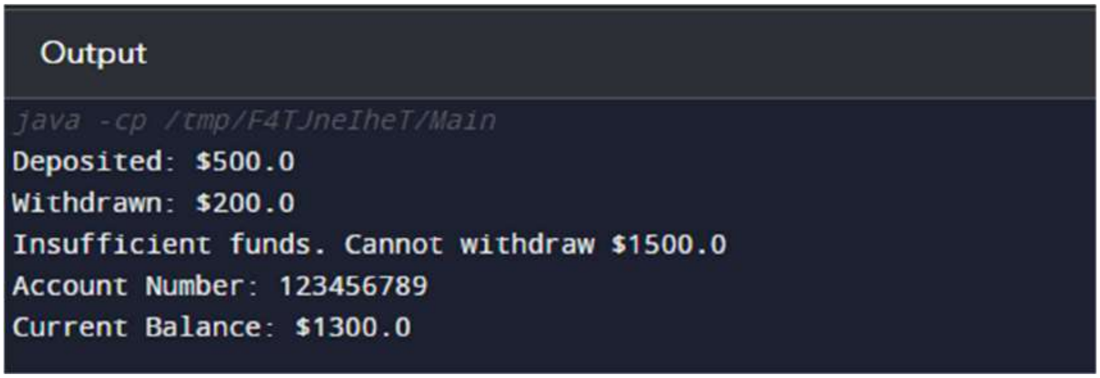
```java
//WAP declaring a Java class called SavingsAccount with members
    ``accountNumber`` and ``Balance``. Provide member functions as
    ``depositAmount ()`` and ``withdrawAmount ()``. If user tries to
     withdraw an amount greater than their balance then throw a user
    -defined exception.
class InsufficientFundsException extends Exception {
    public InsufficientFundsException(String message) {
        super(message);
    }
}
class SavingsAccount {
    private String accountNumber;
    private double balance;
    public SavingsAccount(String accountNumber, double balance) {
        this.accountNumber = accountNumber;
        this.balance = balance;
    }
    public void depositAmount(double amount) {
        balance += amount;
        System.out.println("Deposited: $" + amount);
```



```
    }
    public void withdrawAmount(double amount) throws
        InsufficientFundsException {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawn: $" + amount);
        } else {
            throw new InsufficientFundsException("Insufficient funds
                . Cannot withdraw $" + amount);
        }
    }
    public double getBalance() {
        return balance;
    }
    public String getAccountNumber() {
        return accountNumber;
    }
}
```

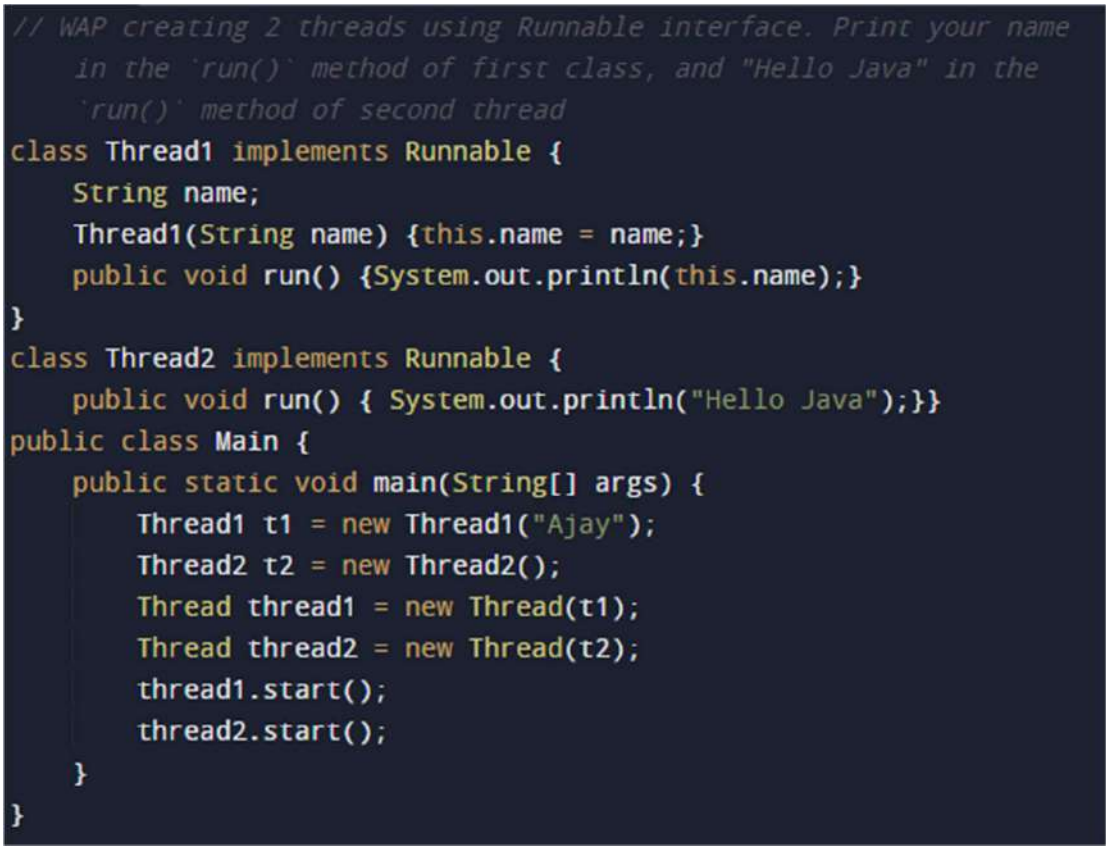


```
public class Main {
    public static void main(String[] args) {
        SavingsAccount account = new SavingsAccount("123456789",
            1000.0);

        try {
            account.depositAmount(500);
            account.withdrawAmount(200);
            account.withdrawAmount(1500);
        } catch (InsufficientFundsException e) {
            System.out.println(e.getMessage());
        }

        System.out.println("Account Number: " + account
            .getAccountNumber());
        System.out.println("Current Balance: $" + account.getBalance
            ());
    }
}
```

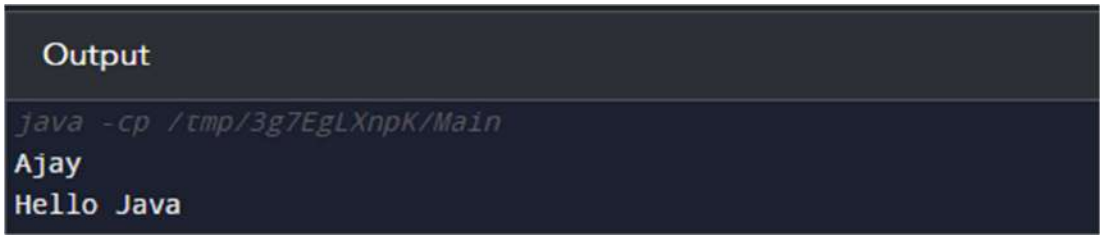


```
java -cp /tmp/F4TJneIheT/Main
Deposited: $500.0
Withdrawn: $200.0
Insufficient funds. Cannot withdraw $1500.0
Account Number: 123456789
Current Balance: $1300.0
```

Ques36: WAP creating 2 threads using Runnable interface. Print your name in ``run ()`` method of first class and "Hello Java" in ``run ()`` method of second thread.

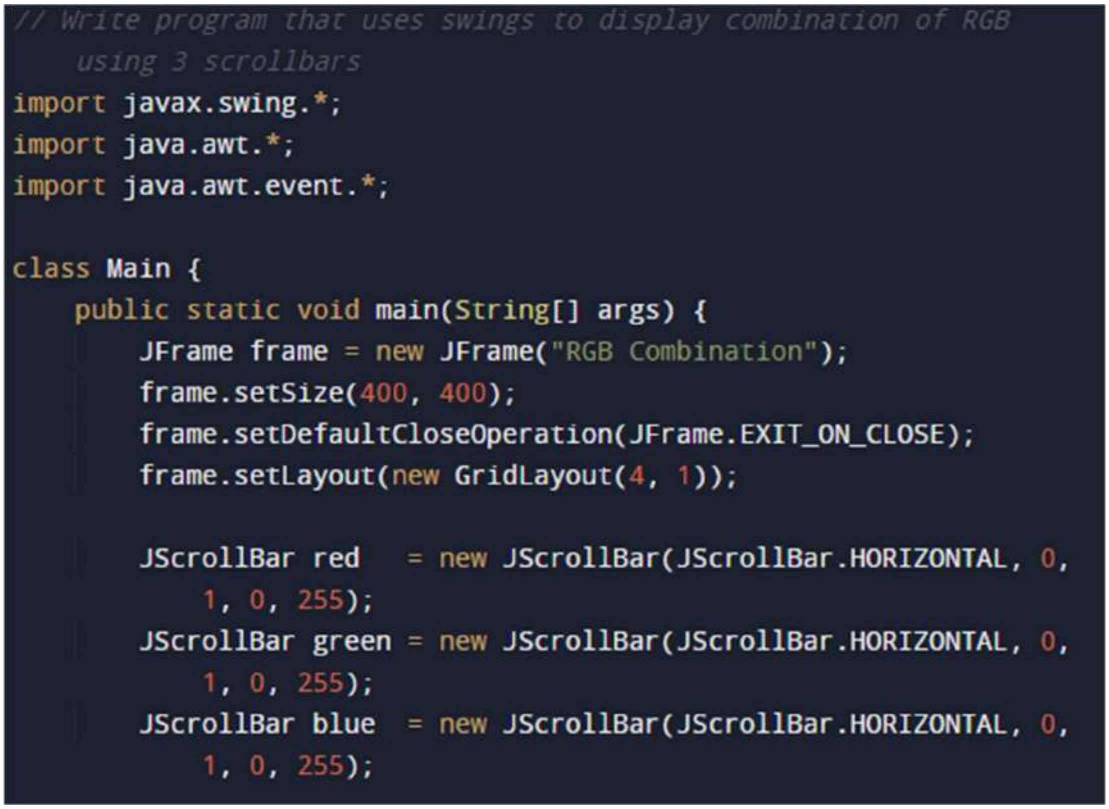


```java
// WAP creating 2 threads using Runnable interface. Print your name
    in the `run()` method of first class, and "Hello Java" in the
    `run()` method of second thread
class Thread1 implements Runnable {
    String name;
    Thread1(String name) {this.name = name;}
    public void run() {System.out.println(this.name);}
}
class Thread2 implements Runnable {
    public void run() { System.out.println("Hello Java");}}
public class Main {
    public static void main(String[] args) {
        Thread1 t1 = new Thread1("Ajay");
        Thread2 t2 = new Thread2();
        Thread thread1 = new Thread(t1);
        Thread thread2 = new Thread(t2);
        thread1.start();
        thread2.start();
    }
}
```

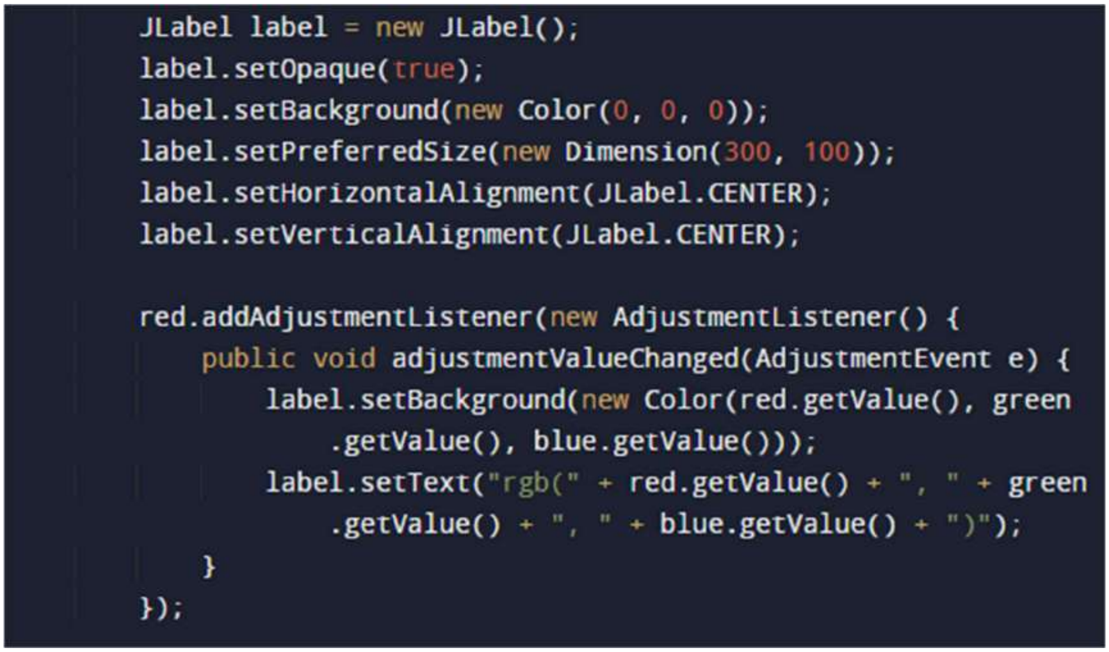


```
java -cp /tmp/3g7EgLXnpK/Main
Ajay
Hello Java
```

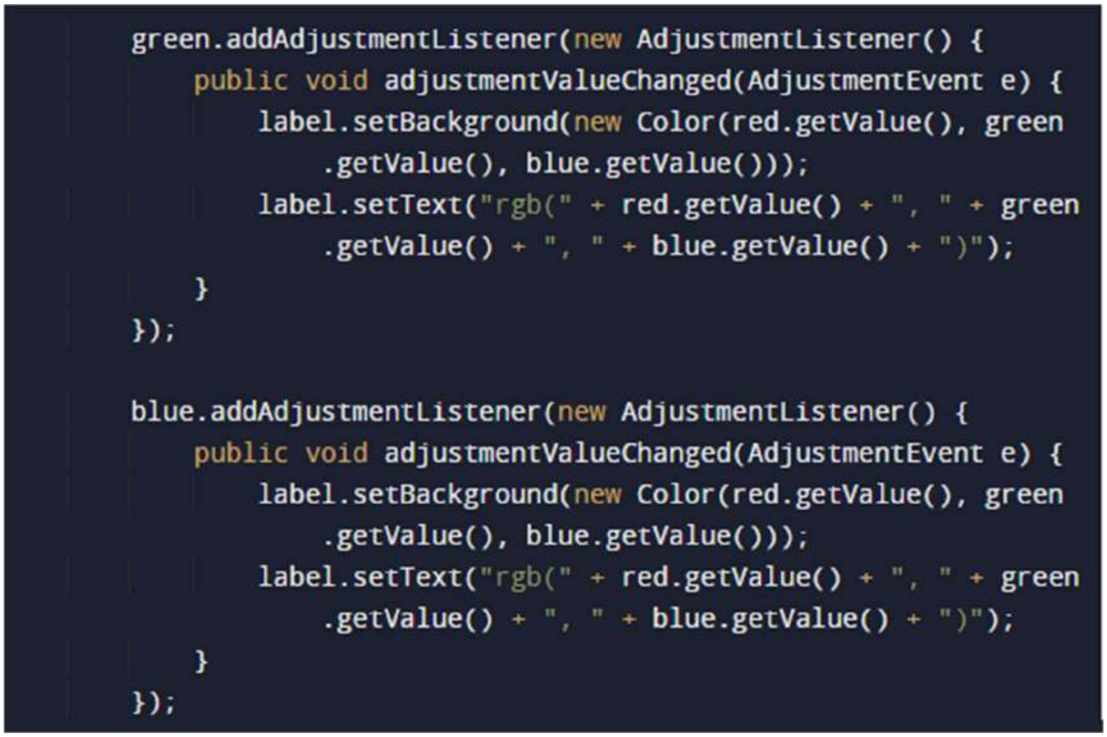
Ques37: Write program that uses swings to display combination of RGB using 3 scrollbars.



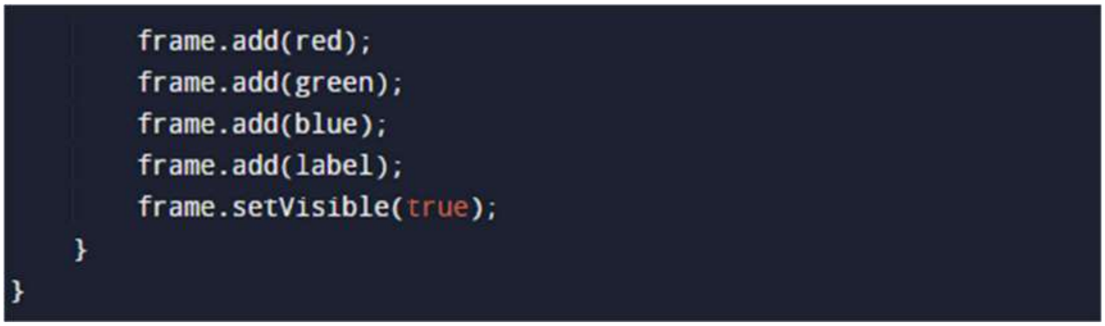
```
// Write program that uses swings to display combination of RGB
    using 3 scrollbars
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class Main {
    public static void main(String[] args) {
        JFrame frame = new JFrame("RGB Combination");
        frame.setSize(400, 400);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new GridLayout(4, 1));

        JScrollBar red   = new JScrollBar(JScrollBar.HORIZONTAL, 0,
            1, 0, 255);
        JScrollBar green = new JScrollBar(JScrollBar.HORIZONTAL, 0,
            1, 0, 255);
        JScrollBar blue  = new JScrollBar(JScrollBar.HORIZONTAL, 0,
            1, 0, 255);
```



```
        JLabel label = new JLabel();
        label.setOpaque(true);
        label.setBackground(new Color(0, 0, 0));
        label.setPreferredSize(new Dimension(300, 100));
        label.setHorizontalAlignment(JLabel.CENTER);
        label.setVerticalAlignment(JLabel.CENTER);

        red.addAdjustmentListener(new AdjustmentListener() {
            public void adjustmentValueChanged(AdjustmentEvent e) {
                label.setBackground(new Color(red.getValue(), green
                    .getValue(), blue.getValue()));
                label.setText("rgb(" + red.getValue() + ", " + green
                    .getValue() + ", " + blue.getValue() + ")");
            }
        });
```

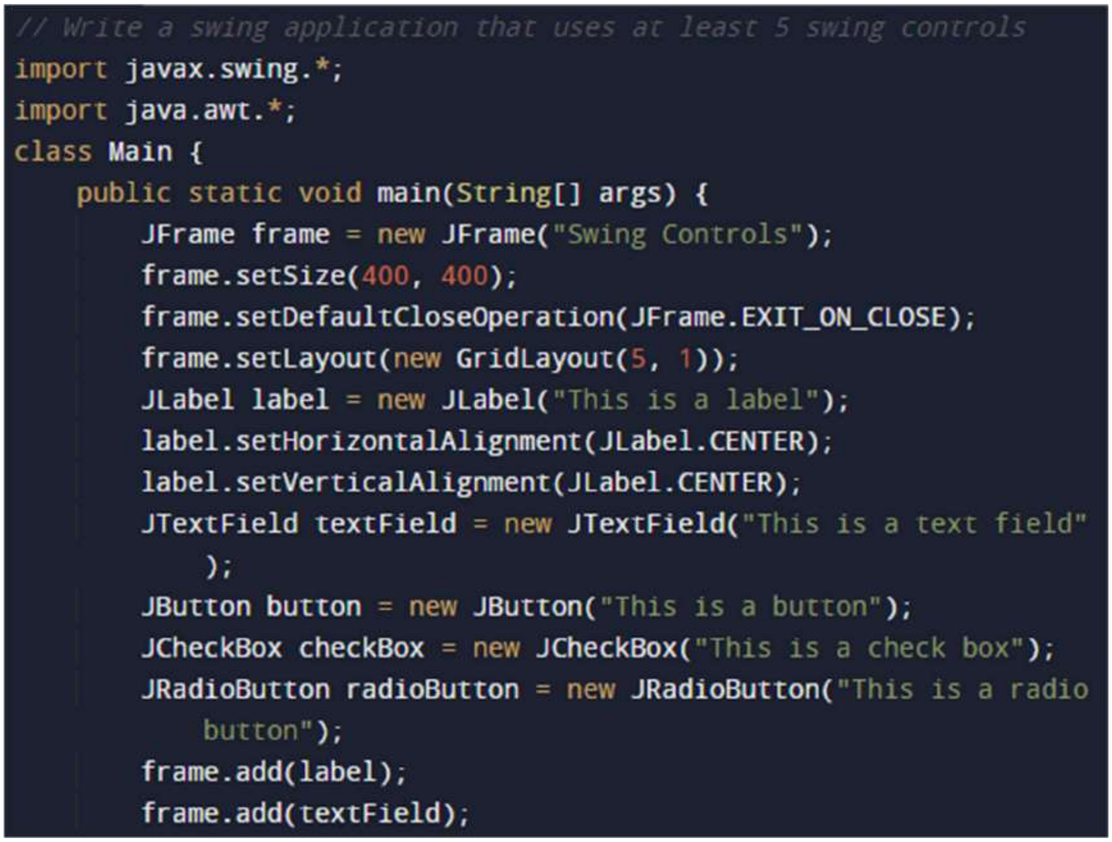


```
        green.addAdjustmentListener(new AdjustmentListener() {
            public void adjustmentValueChanged(AdjustmentEvent e) {
                label.setBackground(new Color(red.getValue(), green
                    .getValue(), blue.getValue()));
                label.setText("rgb(" + red.getValue() + ", " + green
                    .getValue() + ", " + blue.getValue() + ")");
            }
        });

        blue.addAdjustmentListener(new AdjustmentListener() {
            public void adjustmentValueChanged(AdjustmentEvent e) {
                label.setBackground(new Color(red.getValue(), green
                    .getValue(), blue.getValue()));
                label.setText("rgb(" + red.getValue() + ", " + green
                    .getValue() + ", " + blue.getValue() + ")");
            }
        });
```

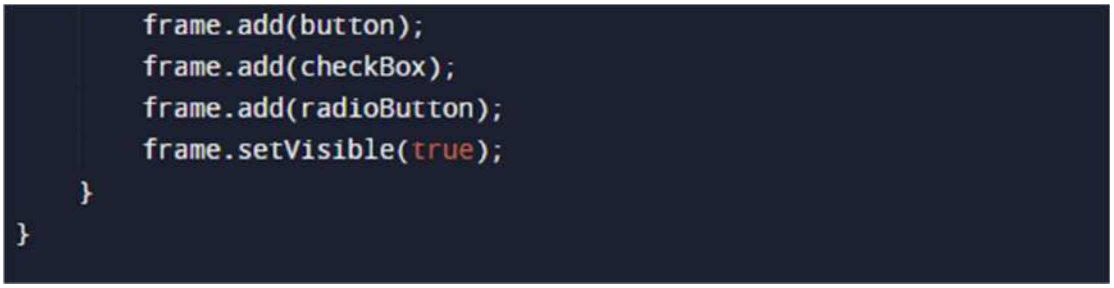


```
        frame.add(red);
        frame.add(green);
        frame.add(blue);
        frame.add(label);
        frame.setVisible(true);
    }
}
```

Ques38: Write a swing application that uses at least 5 swing controls.

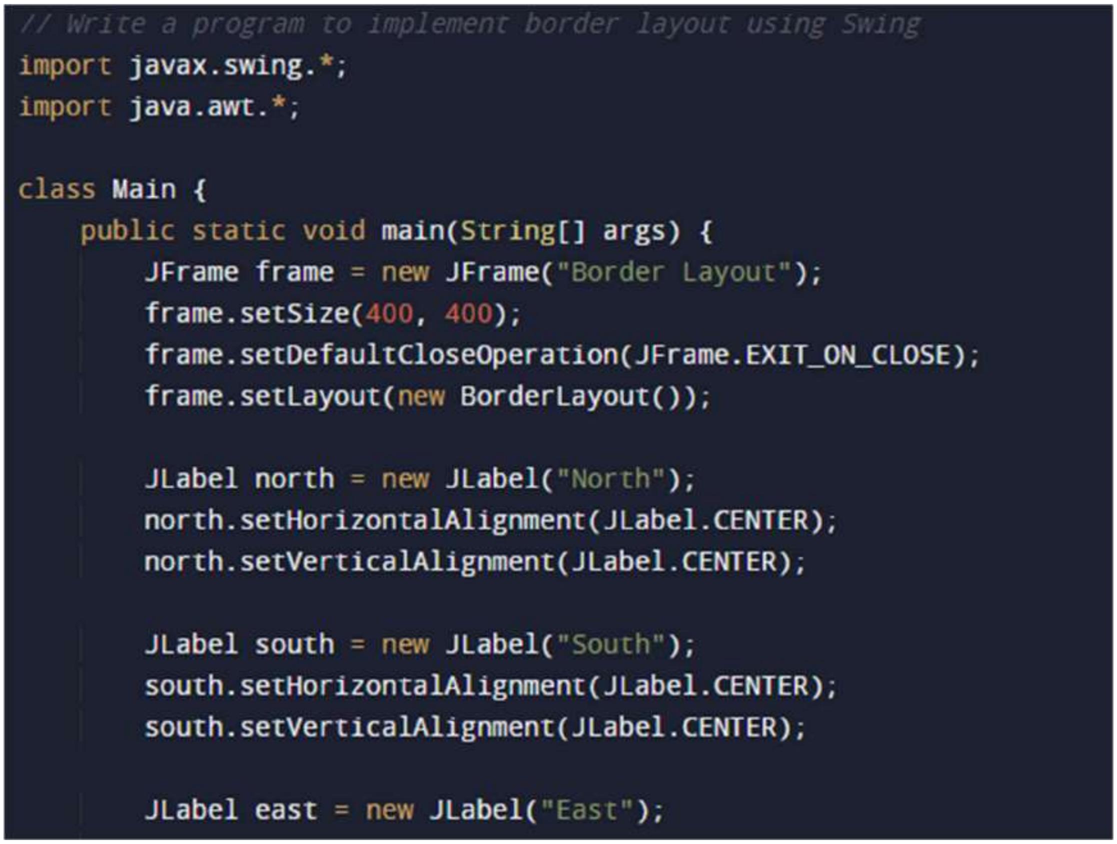


```
// Write a swing application that uses at least 5 swing controls
import javax.swing.*;
import java.awt.*;
class Main {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Swing Controls");
        frame.setSize(400, 400);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new GridLayout(5, 1));
        JLabel label = new JLabel("This is a label");
        label.setHorizontalAlignment(JLabel.CENTER);
        label.setVerticalAlignment(JLabel.CENTER);
        JTextField textField = new JTextField("This is a text field"
            );
        JButton button = new JButton("This is a button");
        JCheckBox checkBox = new JCheckBox("This is a check box");
        JRadioButton radioButton = new JRadioButton("This is a radio
            button");
        frame.add(label);
        frame.add(textField);
```

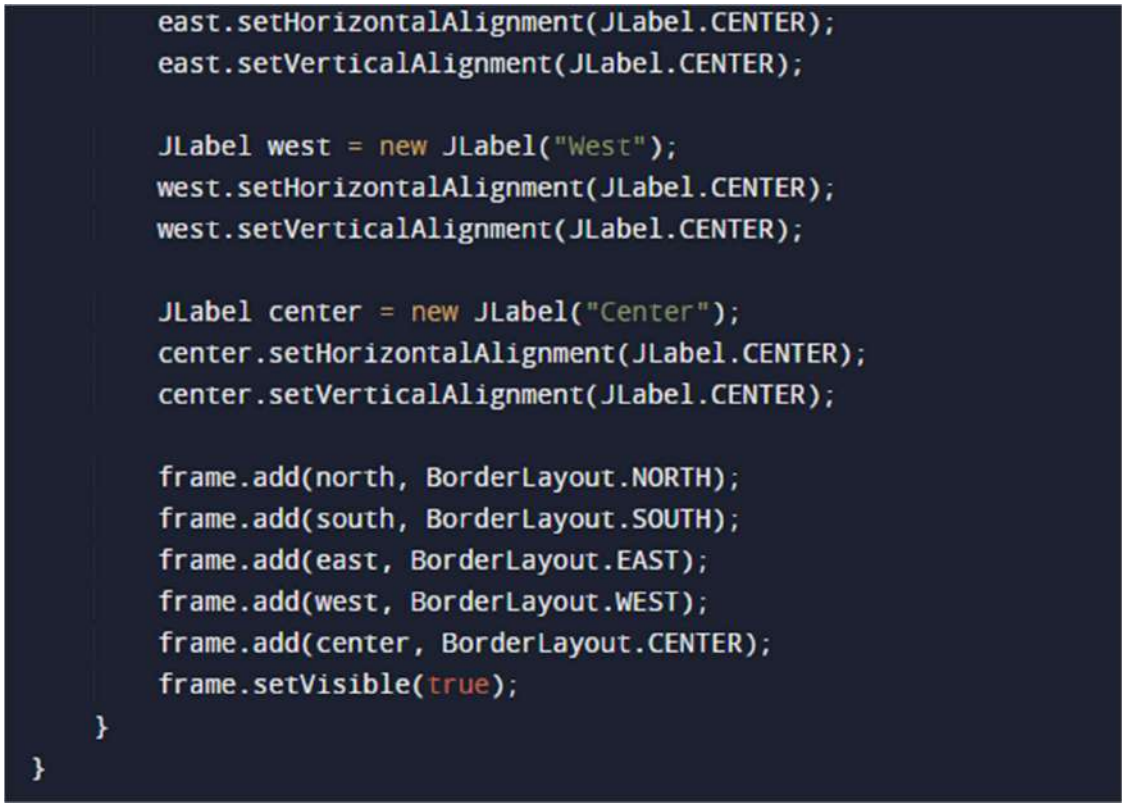


```
        frame.add(button);
        frame.add(checkBox);
        frame.add(radioButton);
        frame.setVisible(true);
    }
}
```

Ques39: Write a program to implement border layout using Swing.

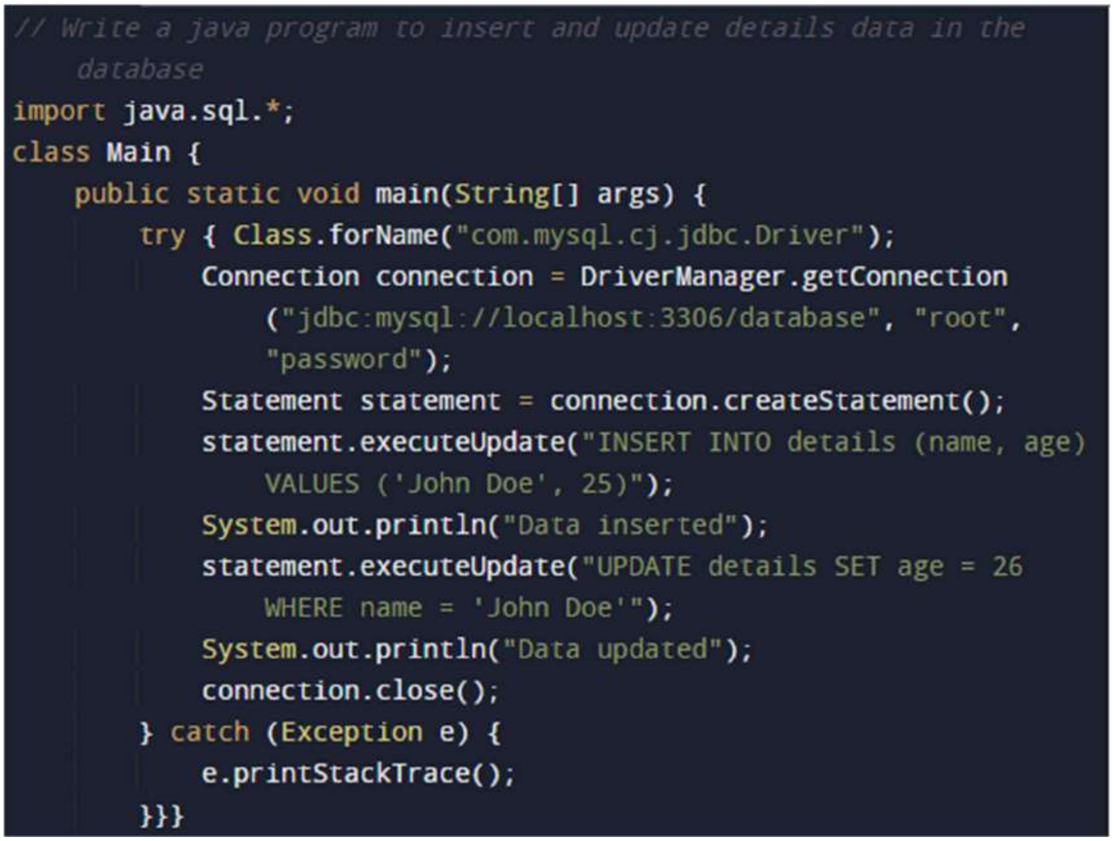


```
// Write a program to implement border layout using Swing
import javax.swing.*;
import java.awt.*;

class Main {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Border Layout");
        frame.setSize(400, 400);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new BorderLayout());

        JLabel north = new JLabel("North");
        north.setHorizontalAlignment(JLabel.CENTER);
        north.setVerticalAlignment(JLabel.CENTER);

        JLabel south = new JLabel("South");
        south.setHorizontalAlignment(JLabel.CENTER);
        south.setVerticalAlignment(JLabel.CENTER);

        JLabel east = new JLabel("East");
```



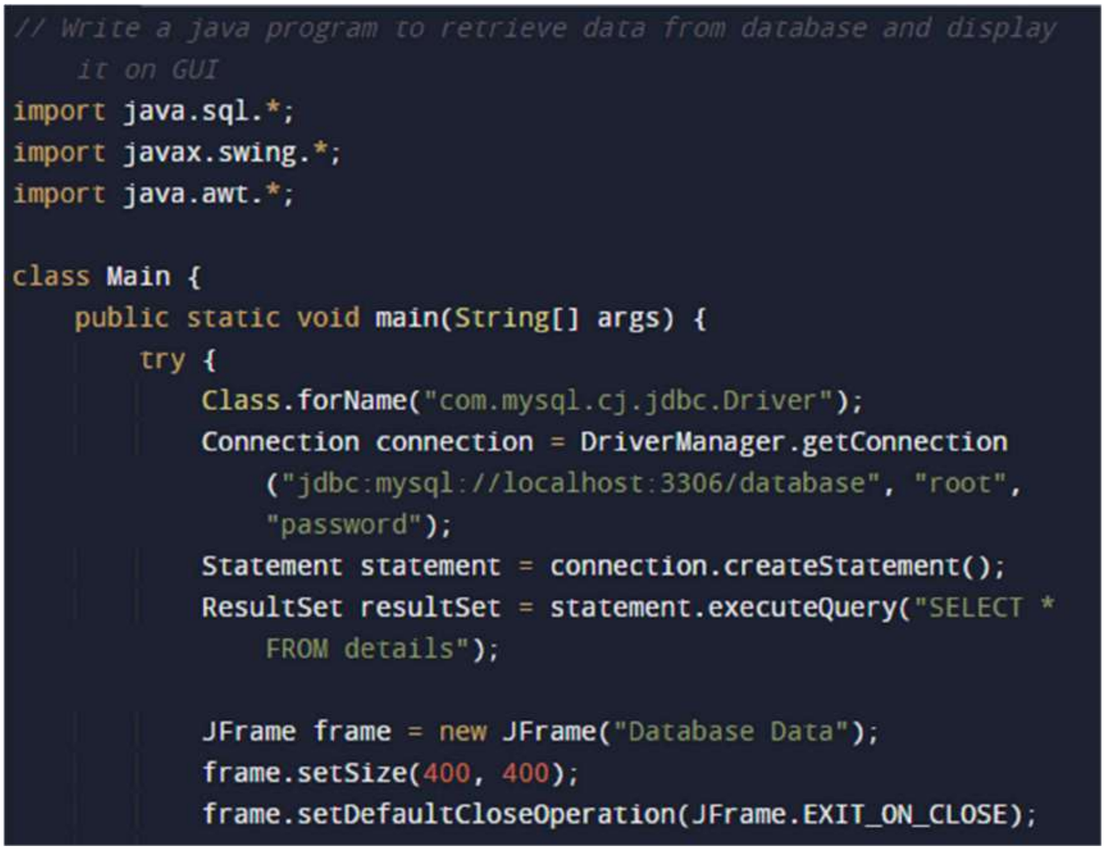
```
        east.setHorizontalAlignment(JLabel.CENTER);
        east.setVerticalAlignment(JLabel.CENTER);

        JLabel west = new JLabel("West");
        west.setHorizontalAlignment(JLabel.CENTER);
        west.setVerticalAlignment(JLabel.CENTER);

        JLabel center = new JLabel("Center");
        center.setHorizontalAlignment(JLabel.CENTER);
        center.setVerticalAlignment(JLabel.CENTER);

        frame.add(north, BorderLayout.NORTH);
        frame.add(south, BorderLayout.SOUTH);
        frame.add(east, BorderLayout.EAST);
        frame.add(west, BorderLayout.WEST);
        frame.add(center, BorderLayout.CENTER);
        frame.setVisible(true);
    }
}
```

Ques40: Write a java program to insert and update details data in the database.

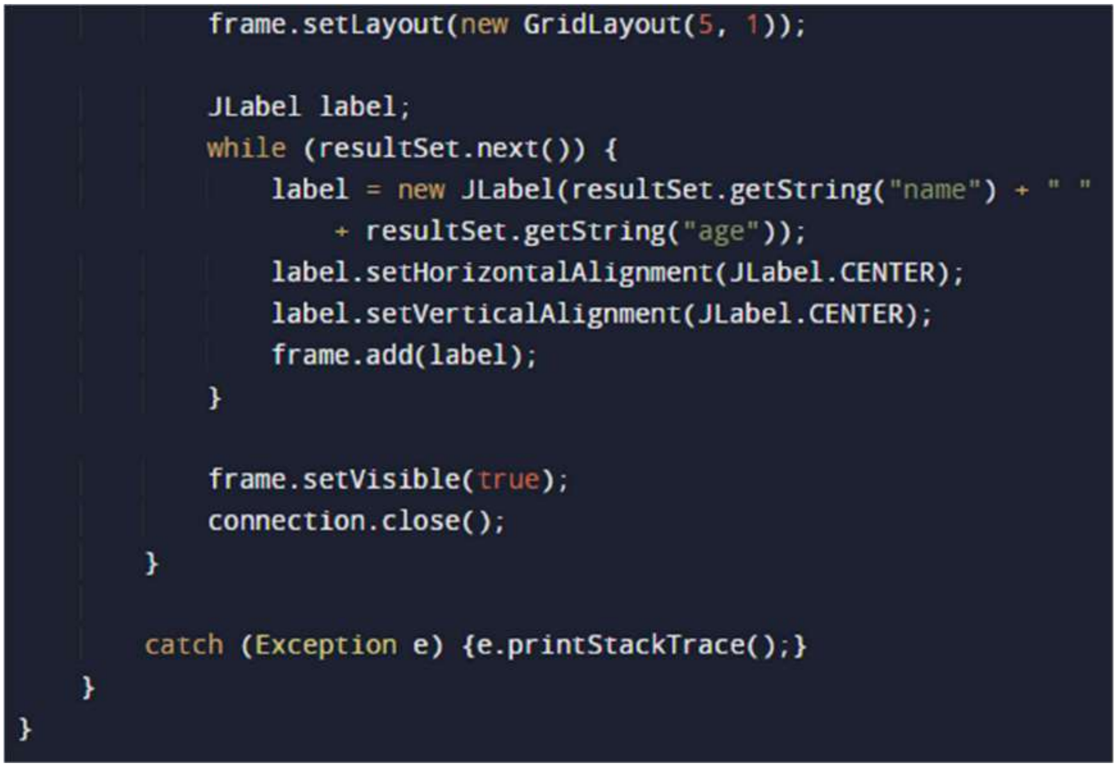


```
// Write a java program to insert and update details data in the
    database
import java.sql.*;
class Main {
    public static void main(String[] args) {
        try { Class.forName("com.mysql.cj.jdbc.Driver");
            Connection connection = DriverManager.getConnection
                ("jdbc:mysql://localhost:3306/database", "root",
                "password");
            Statement statement = connection.createStatement();
            statement.executeUpdate("INSERT INTO details (name, age)
                VALUES ('John Doe', 25)");
            System.out.println("Data inserted");
            statement.executeUpdate("UPDATE details SET age = 26
                WHERE name = 'John Doe'");
            System.out.println("Data updated");
            connection.close();
        } catch (Exception e) {
            e.printStackTrace();
        }}}
```

Ques41: Write a java program to retrieve data from database and display it on GUI.



```
// Write a java program to retrieve data from database and display
    it on GUI
import java.sql.*;
import javax.swing.*;
import java.awt.*;

class Main {
    public static void main(String[] args) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection connection = DriverManager.getConnection
                ("jdbc:mysql://localhost:3306/database", "root",
                "password");
            Statement statement = connection.createStatement();
            ResultSet resultSet = statement.executeQuery("SELECT *
                FROM details");

            JFrame frame = new JFrame("Database Data");
            frame.setSize(400, 400);
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```



```
            frame.setLayout(new GridLayout(5, 1));

            JLabel label;
            while (resultSet.next()) {
                label = new JLabel(resultSet.getString("name") + " "
                    + resultSet.getString("age"));
                label.setHorizontalAlignment(JLabel.CENTER);
                label.setVerticalAlignment(JLabel.CENTER);
                frame.add(label);
            }

            frame.setVisible(true);
            connection.close();
        }

        catch (Exception e) {e.printStackTrace();}
    }
}
```