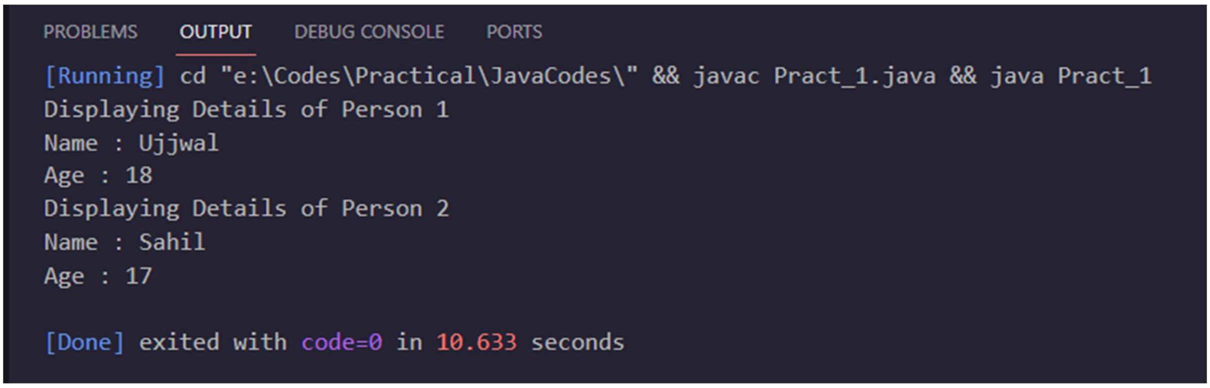
1.Write a program in java to implement the classes and object.

```
class Person

{
        String name;

        int age;

     public Person(String name, int age){
            this.name = name;

            this.age = age;

     }

// Creating Method in Person Class

public void out()

{

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

          System.out.println("Age : " + age);

          }

}

public class Pract_1

{

public static void main(String args[])

         {

            // Creating Objects of Person Class

            Person obj1 = new Person ("Ujjwal",18);

            Person obj2 = new Person("Sahil",17);

           // Caliing the method of Person class with help of Object

            System.out.println("Displaying Details of Person 1 " );

            obj1.out();

            System.out.println("Displaying Details of Person 2 " );

            obj2.out();

         }

}
```

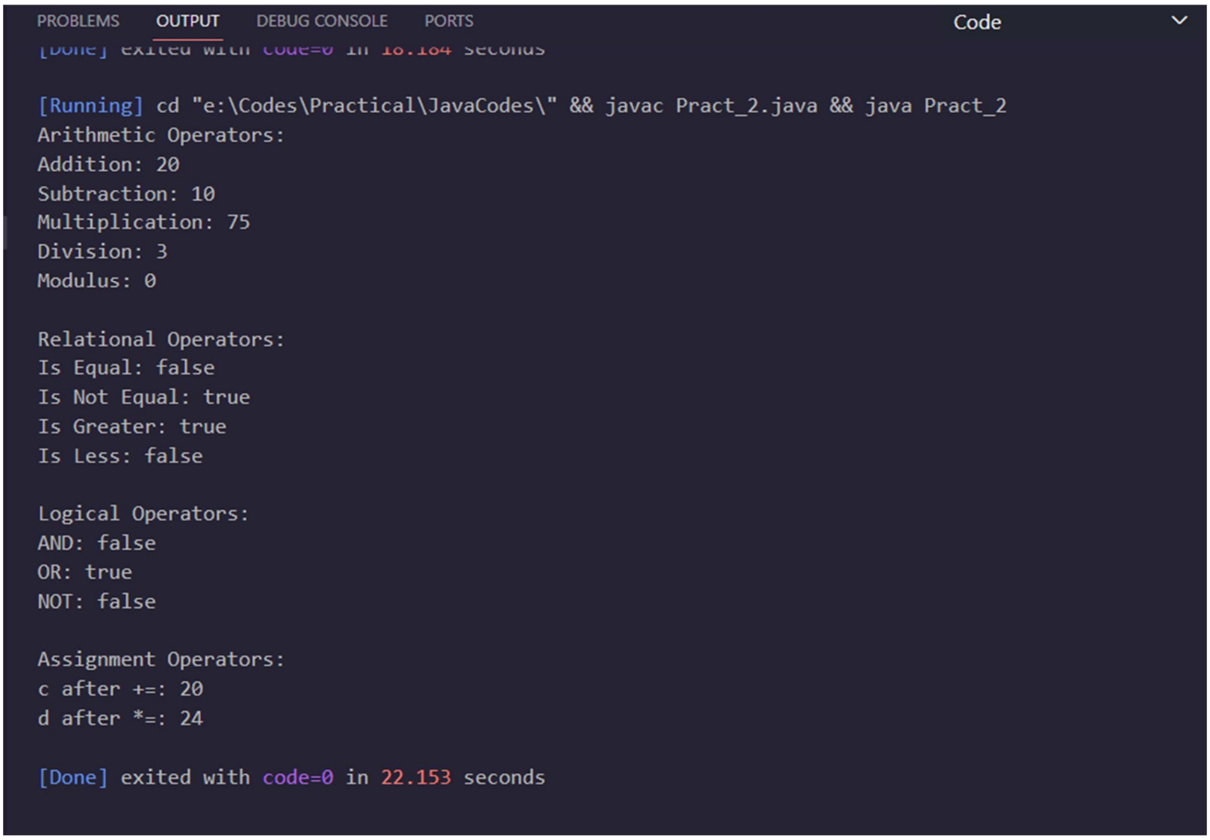
Output :



```
[Running] cd "e:\Codes\Practical\JavaCodes\" && javac Pract_1.java && java Pract_1
Displaying Details of Person 1
Name : Ujjwal
Age : 18
Displaying Details of Person 2
Name : Sahil
Age : 17

[Done] exited with code=0 in 10.633 seconds
```

2. Write a Program on the use of operators in Java.

```
public class Pract_2 {
public static void main(String[] args) {
// Arithmetic operators
int a = 15, b = 5;
int addition = a + b;
int subtraction = a - b;
int multiplication = a * b;
int division = a / b;
int modulus = a % b;
System.out.println("Arithmetic Operators:");
System.out.println("Addition: " + addition);
System.out.println("Subtraction: " + subtraction);
System.out.println("Multiplication: " + multiplication);
System.out.println("Division: " + division);
System.out.println("Modulus: " + modulus);
// Relational operators
boolean isEqual = (a == b);
boolean isNotEqual = (a != b);
boolean isGreater = (a > b);
boolean isLess = (a < b);
System.out.println("\nRelational Operators:");
System.out.println("Is Equal: " + isEqual);
System.out.println("Is Not Equal: " + isNotEqual);
System.out.println("Is Greater: " + isGreater);
System.out.println("Is Less: " + isLess);
// Logical operators
boolean x = true, y = false;
boolean andResult = x && y;
boolean orResult = x || y;
boolean notResult = !x;

System.out.println("\nLogical Operators:");

System.out.println("AND: " + andResult);

System.out.println("OR: " + orResult);

System.out.println("NOT: " + notResult);

// Assignment operators

int c = 15;

c += 5; // Equivalent to c = c + 5

int d = 8;

d *= 3; // Equivalent to d = d * 3

System.out.println("\nAssignment Operators:");

System.out.println("c after +=: " + c);

System.out.println("d after *=: " + d);

}

}
```



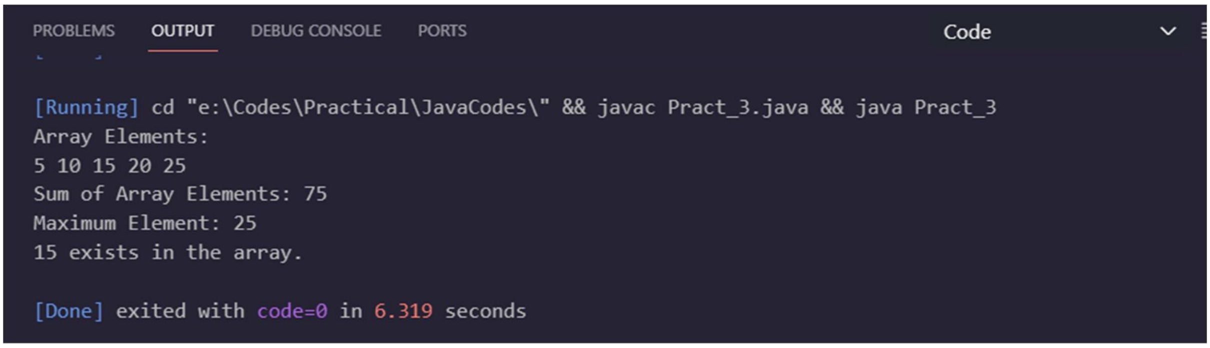
```
[Done] exited with code=0 in 18.184 seconds

[Running] cd "e:\Codes\Practical\JavaCodes\" && javac Pract_2.java && java Pract_2
Arithmetic Operators:
Addition: 20
Subtraction: 10
Multiplication: 75
Division: 3
Modulus: 0

Relational Operators:
Is Equal: false
Is Not Equal: true
Is Greater: true
Is Less: false

Logical Operators:
AND: false
OR: true
NOT: false

Assignment Operators:
c after +=: 20
d after *=: 24

[Done] exited with code=0 in 22.153 seconds
```

3.Write a Program of using Arrays and control Statements.

```
class Pract_3 {
public static void main(String[] args) {
// Declare and initialize an array of integers
int[] numbers = { 5, 10, 15, 20, 25 };
// Loop through the array and print each element
System.out.println("Array Elements:");
for (int i = 0; i < numbers.length; i++) {
System.out.print(numbers[i] + " ");
}
System.out.println(); // Print a newline
// Calculate the sum of array elements
int sum = 0;
for (int i = 0; i < numbers.length; i++) {
sum += numbers[i];
}
System.out.println("Sum of Array Elements: " + sum);
// Find the maximum element in the array
int max = numbers[0];
for (int i = 1; i < numbers.length; i++) {
if (numbers[i] > max) {
max = numbers[i];
}
}
System.out.println("Maximum Element: " + max);
// Use control statements to check if an element exists in the array
int searchValue = 15;
boolean found = false;
for (int i = 0; i < numbers.length; i++) {
if (numbers[i] == searchValue) {
```

```
found = true;
break; // Exit the loop when found
}
}
if (found) {
System.out.println(searchValue + " exists in the array.");
} else {
System.out.println(searchValue + " does not exist in the array.");
}
}
}
```

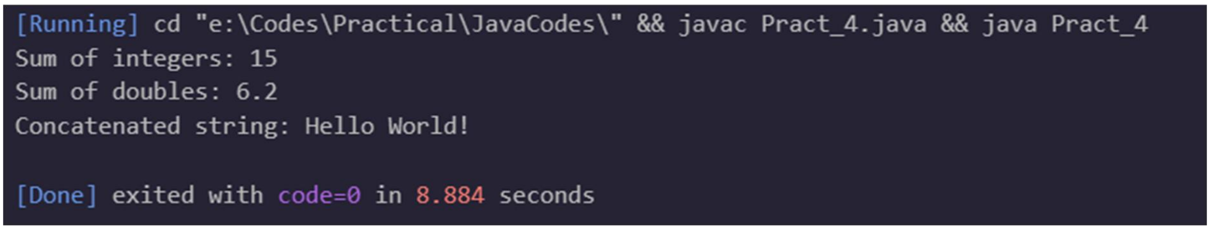
Output:



```
[Running] cd "e:\Codes\Practical\JavaCodes\" && javac Pract_3.java && java Pract_3
Array Elements:
5 10 15 20 25
Sum of Array Elements: 75
Maximum Element: 25
15 exists in the array.

[Done] exited with code=0 in 6.319 seconds
```

4.Write a program in Java to implement method Overloading.

```
class Calculator {
int add(int a, int b) {
return a + b;
}
double add(double a, double b) {
return a + b;
}
String add(String a, String b) {
return a + b;
}
}
class Pract_4 {
public static void main(String[] args) {
Calculator cal = new Calculator();
int a = cal.add(5, 10);
System.out.println("Sum of integers: " + a);
double b = cal.add(3.5, 2.7);
System.out.println("Sum of doubles: " + b);
String c = cal.add("Hello", " World!");
System.out.println("Concatenated string: " + c);
}
}
```



```
[Running] cd "e:\Codes\Practical\JavaCodes\" && javac Pract_4.java && java Pract_4
Sum of integers: 15
Sum of doubles: 6.2
Concatenated string: Hello World!

[Done] exited with code=0 in 8.884 seconds
```

5.Write a program in Java to implement Inheritance.

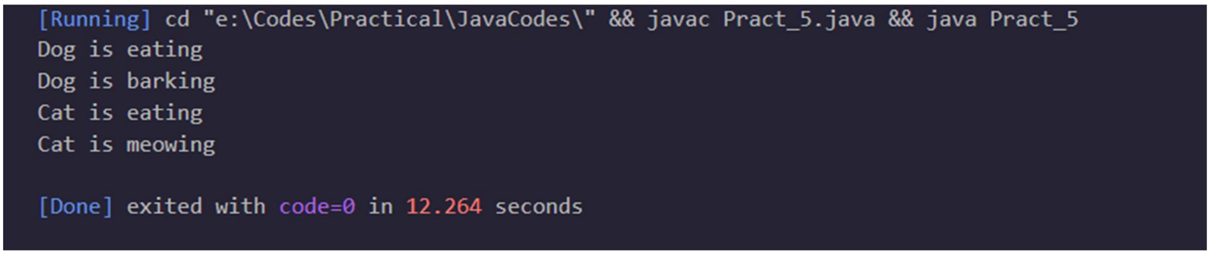
```
class Animal {
void eat() {
System.out.println("Animal is eating");
}
}
class Dog extends Animal {
void eat() {
System.out.println("Dog is eating");
}
void bark() {
System.out.println("Dog is barking");
}
}
class Cat extends Animal {
void eat() {
System.out.println("Cat is eating");
}
void meow() {
System.out.println("Cat is meowing");
}
}
class Pract_5 {
public static void main(String[] args) {
// Creating Dog Class Object
Dog myDog = new Dog();
myDog.eat();
myDog.bark();
// Creating Cat Class Object
Cat myCat = new Cat();
```

myCat.eat();

myCat.meow();

}

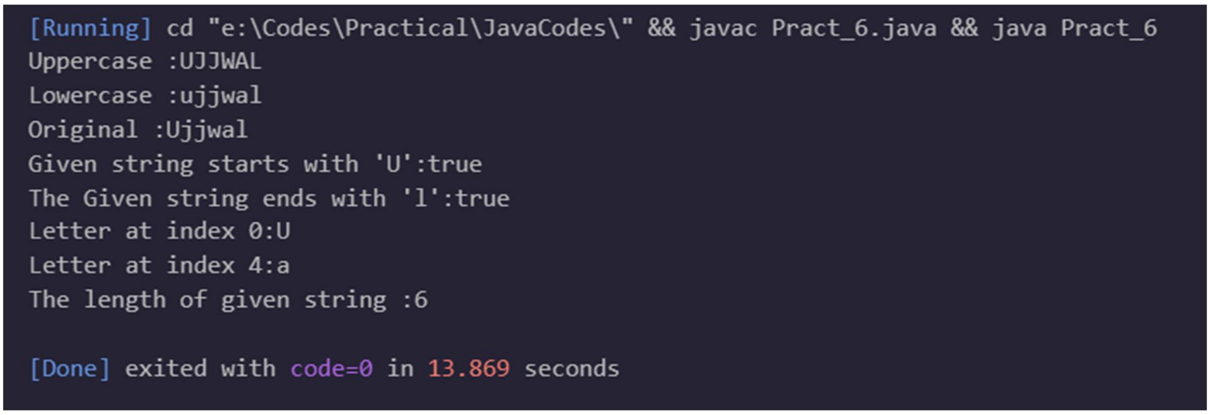
}



```
[Running] cd "e:\Codes\Practical\JavaCodes\" && javac Pract_5.java && java Pract_5
Dog is eating
Dog is barking
Cat is eating
Cat is meowing

[Done] exited with code=0 in 12.264 seconds
```

6. Write a program to use the String class in java.

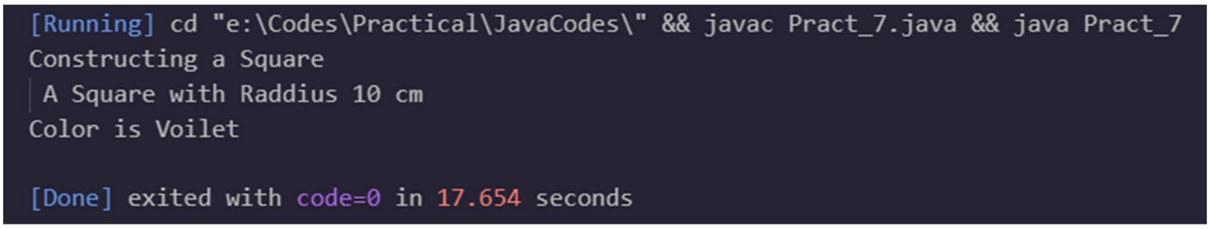
```
public class Pract_6 {

public static void main(String args[]) {

String x="Ujjwal";


System.out.print("Uppercase :");

System.out.println(x.toUpperCase());

System.out.print("Lowercase :");

System.out.println(x.toLowerCase());

System.out.print("Original :");

System.out.println(x);

System.out.print("Given string starts with 'U':");

System.out.println(x.startsWith("U"));

System.out.print("The Given string ends with 'l':");

System.out.println(x.endsWith("l"));

System.out.print("Letter at index 0:");

System.out.println(x.charAt(0));

System.out.print("Letter at index 4:");

System.out.println(x.charAt(4));

System.out.print("The length of given string :");

System.out.println(x.length());

}

}
```



```
[Running] cd "e:\Codes\Practical\JavaCodes\" && javac Pract_6.java && java Pract_6
Uppercase :UJJWAL
Lowercase :ujjwal
Original :Ujjwal
Given string starts with 'U':true
The Given string ends with 'l':true
Letter at index 0:U
Letter at index 4:a
The length of given string :6

[Done] exited with code=0 in 13.869 seconds
```

7.Write a program for implement the Abstract Class.

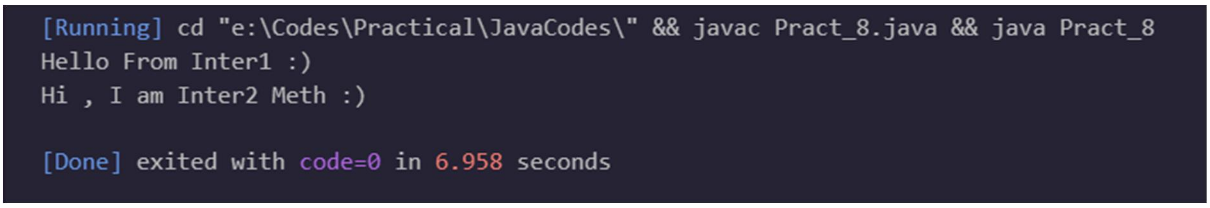
```
abstract class Shape {
abstract void draw();
abstract void print();
abstract void color();
}
class Square extends Shape {
void draw() {
System.out.println("Constructing a Square");
}
void print() {
System.out.println(" A Square with Raddius 10 cm");
}
void color(){
 System.out.println("Color is Voilet");
}
}
class Pract_7{
public static void main(String[] args) {
    Square obj = new Square();
     obj.draw();
     obj.print();
     obj.color();
}
}
```



```
[Running] cd "e:\Codes\Practical\JavaCodes\" && javac Pract_7.java && java Pract_7
Constructing a Square
 A Square with Raddius 10 cm
Color is Voilet

[Done] exited with code=0 in 17.654 seconds
```

8. Write a program for implementation of Interfaces Java.

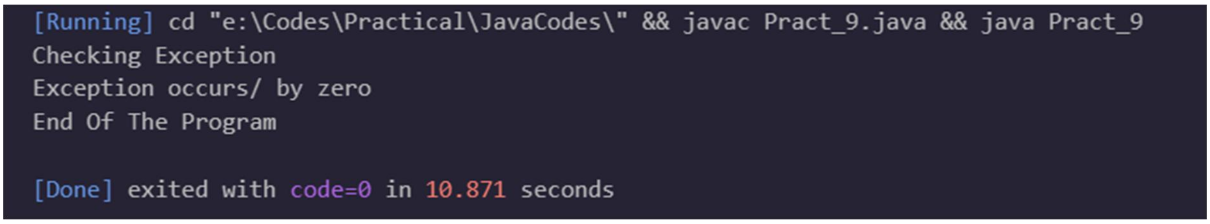
```
interface Inter1 {
void meth1();
}
interface Inter2 {
void meth2();
}
class InterfaceClass implements Inter1, Inter2 {
public void meth1() {
System.out.println("Hello From Inter1 :) ");
}
public void meth2() {
System.out.println("Hi , I am Inter2 Meth :)");
}
}
class Pract_8{
 public static void main(String[] args) {
 InterfaceClass obj = new InterfaceClass();
 obj.meth1();
 obj.meth2();
 }
}
```



```
[Running] cd "e:\Codes\Practical\JavaCodes\" && javac Pract_8.java && java Pract_8
Hello From Inter1 :)
Hi , I am Inter2 Meth :)

[Done] exited with code=0 in 6.958 seconds
```

9.Write a Program in Java to use the Exception Handling (Try, Catch, Finally Blocks).

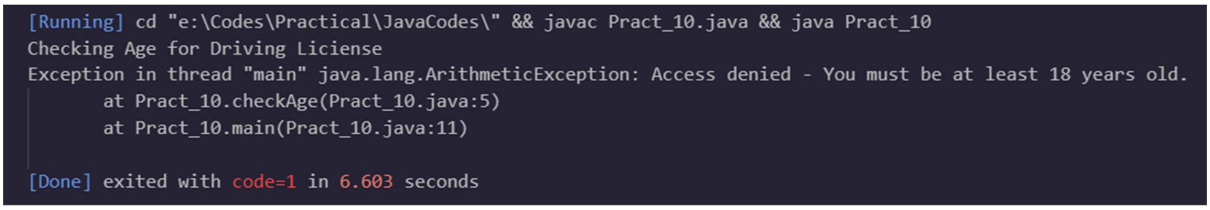
```
class Pract_9 {
 public static void main(String []args){
 System.out.println("Checking Exception ");
 try{
 int a = 2;
 int b = 0;
 System.out.println(a/b);
 }catch(Exception e){

System.out.println("Exception occurs"+e.getMessage());

 }finally{

System.out.println("End Of The Program");
            }
}
}
```



```
[Running] cd "e:\Codes\Practical\JavaCodes\" && javac Pract_9.java && java Pract_9
Checking Exception
Exception occurs/ by zero
End Of The Program

[Done] exited with code=0 in 10.871 seconds
```

10. Write a program in java to use the keywords throw and throws in Exceptional Handling.

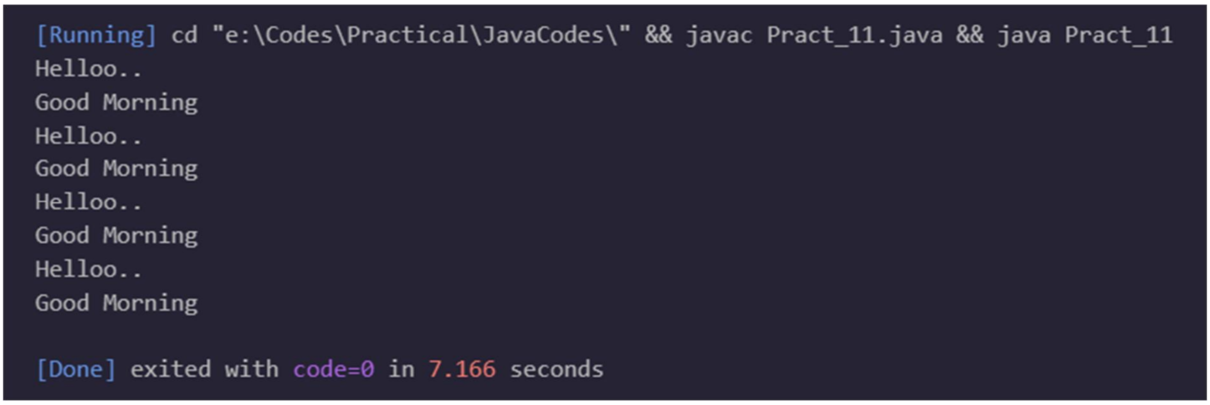
```
class Pract_10 {
 static void checkAge(int age) throws ArithmeticException {
   System.out.println("Checking Age for Driving Liciense ");
 if (age <= 18) {
 throw new ArithmeticException("Access denied - You must be at least 18 years old.");
 } else {
 System.out.println("Access granted - You are old enough!");
 }
 }
 public static void main(String[] args) {
 checkAge(17);
 }
 }
```



```
[Running] cd "e:\Codes\Practical\JavaCodes\" && javac Pract_10.java && java Pract_10
Checking Age for Driving Liciense
Exception in thread "main" java.lang.ArithmeticException: Access denied - You must be at least 18 years old.
	at Pract_10.checkAge(Pract_10.java:5)
	at Pract_10.main(Pract_10.java:11)

[Done] exited with code=1 in 6.603 seconds
```

11.Write a program to Implements Thread Methods of Multithreading in Java.

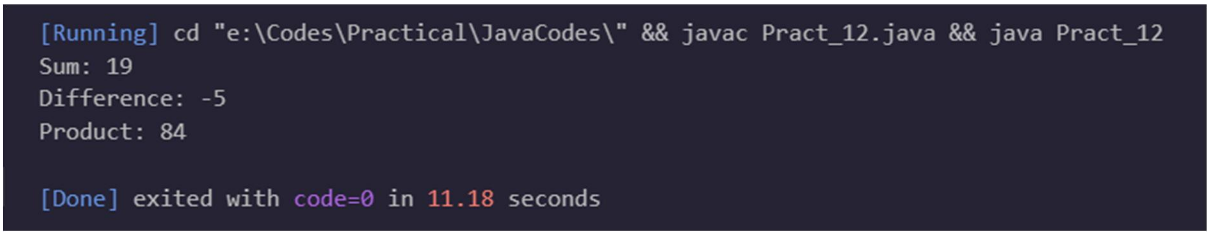
```
class Greet extends Thread {
public void run() {
for (int i = 1; i < 3; i++) {
   System.out.println("Helloo..");
System.out.println( "Good Morning");


}
}
}
public class Pract_11 {
public static void main(String[] args) {
Greet thread1 = new Greet();
Greet thread2 = new Greet();
thread1.start();
thread2.start();
}
}
```



```
[Running] cd "e:\Codes\Practical\JavaCodes\" && javac Pract_11.java && java Pract_11
Helloo..
Good Morning
Helloo..
Good Morning
Helloo..
Good Morning
Helloo..
Good Morning

[Done] exited with code=0 in 7.166 seconds
```

12.Write a program on the use of Utility classes.

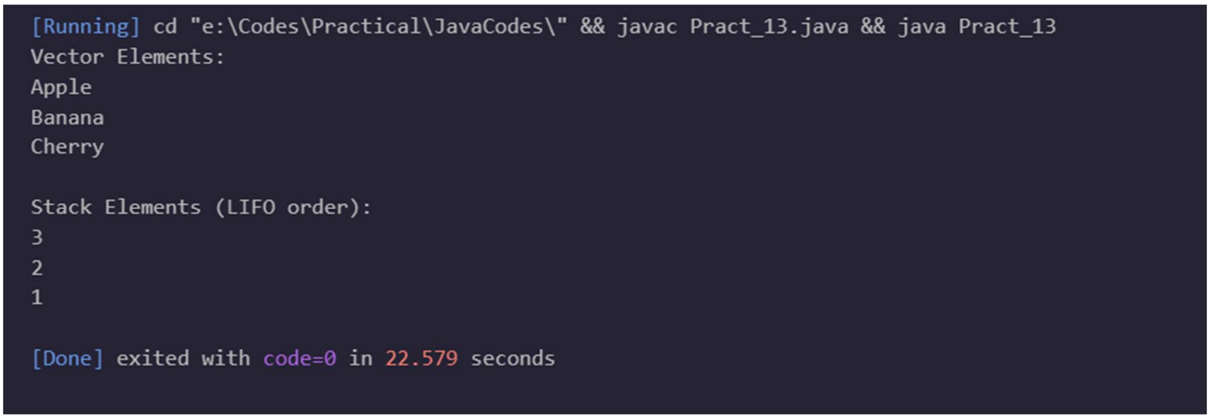
```
class MathUtils {

 private MathUtils() {

 }

 public static int add(int a, int b) {

 return a + b;

 }

 public static int subtract(int a, int b) {

 return a - b;

 }

 public static int multiply(int a, int b) {

 return a * b;

 }

}

class Pract_12 {

 public static void main(String[] args) {

 int num1 = 7;

 int num2 = 12;

 int sum = MathUtils.add(num1, num2);

 int difference = MathUtils.subtract(num1, num2);

 int product = MathUtils.multiply(num1, num2);

 System.out.println("Sum: " + sum);

 System.out.println("Difference: " + difference);

 System.out.println("Product: " + product);

 }

}
```



```
[Running] cd "e:\Codes\Practical\JavaCodes\" && javac Pract_12.java && java Pract_12
Sum: 19
Difference: -5
Product: 84

[Done] exited with code=0 in 11.18 seconds
```

13.Write a program to implements Vector class and stack classes in Legacy classes in Java.

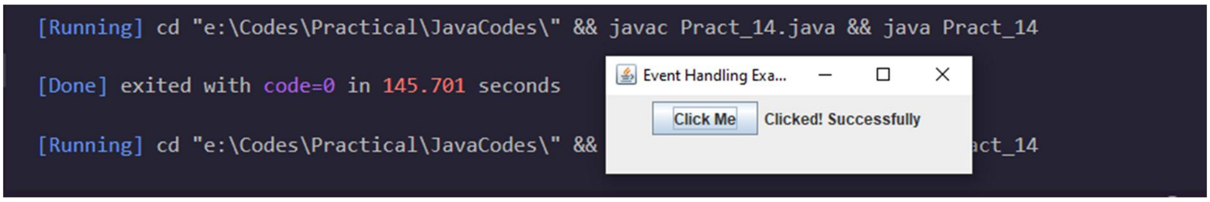
```
import java.util.Vector;
import java.util.Stack;

class Pract_13 {
 public static void main(String[] args) {

 // Using the Vector class
 Vector<String> vector = new Vector<>();

 // Adding elements to the Vector
 vector.add("Apple");
 vector.add("Banana");
 vector.add("Cherry");

 // Accessing elements in the Vector
 System.out.println("Vector Elements:");
 for (String fruit : vector) {
 System.out.println(fruit);
 }
 // Using the Stack class
 Stack<Integer> stack = new Stack<>();

 // Pushing elements onto the stack
 stack.push(1);
 stack.push(2);
 stack.push(3);
 // Popping elements from the stack
 System.out.println("\nStack Elements (LIFO order):");
 while (!stack.isEmpty()) {
 System.out.println(stack.pop());
 }
 }
}
```



```
[Running] cd "e:\Codes\Practical\JavaCodes\" && javac Pract_13.java && java Pract_13
Vector Elements:
Apple
Banana
Cherry

Stack Elements (LIFO order):
3
2
1

[Done] exited with code=0 in 22.579 seconds
```

14. Write a Program on using Event Handling.

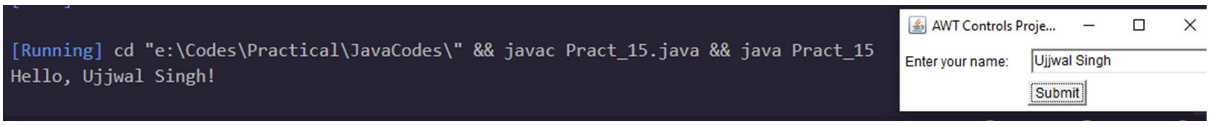
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

class Pract_14 {
 public static void main(String[] args) {
 // Create a JFrame (a window)
 JFrame frame = new JFrame("Event Handling Example");
 // Create a button
 JButton button = new JButton("Click Me");
 // Create a label to display the message
 JLabel label = new JLabel("");
 // Add an ActionListener to the button
 button.addActionListener(new ActionListener() {
 @Override
 public void actionPerformed(ActionEvent e) {
 label.setText("Clicked! Successfully ");
 }
 });
 // Set the layout manager for the frame
 frame.setLayout(new FlowLayout());
 // Add the button and label to the frame
 frame.add(button);
 frame.add(label);
 // Set the frame size and close operation
 frame.setSize(300, 100);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Make the frame visible

frame.setVisible(true);

}

}
```



```
[Running] cd "e:\Codes\Practical\JavaCodes\" && javac Pract_14.java && java Pract_14

[Done] exited with code=0 in 145.701 seconds

[Running] cd "e:\Codes\Practical\JavaCodes\" && 
act_14
```

15. Write a program on the use of AWT Controls in Java.

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

class Pract_15 {
public static void main(String[] args) {
// Create a Frame (window)
Frame frame = new Frame("AWT Controls Project -01 ");
// Create a Label
Label label = new Label("Enter your name: ");
// Create a Text Field
TextField textField = new TextField(20);
 // Create a Button
Button button = new Button("Submit");
// Create an ActionListener for the Button
button.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
String name = textField.getText();
System.out.println("Hello, " + name + "!");
}
});
 // Add components to the Frame
frame.setLayout(new FlowLayout());
frame.add(label);
frame.add(textField);
frame.add(button);
// Set the Frame size and make it visible
frame.setSize(300, 100);
frame.setVisible(true);
// Add a WindowListener to handle window closing
frame.addWindowListener(new WindowAdapter() {
public void windowClosing(WindowEvent e) {
 System.exit(0);
}
});
}
}
```



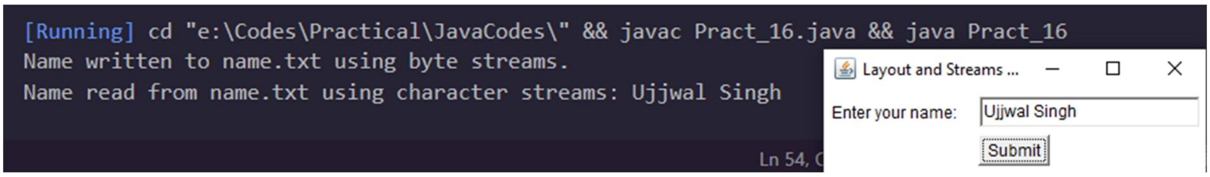
```
[Running] cd "e:\Codes\Practical\JavaCodes\" && javac Pract_15.java && java Pract_15
Hello, Ujjwal Singh!
```

Enter your name: Ujjwal Singh

16. Write a Program on the use of Layout Managers, Byte and Character Streams in Java.

```
import java.awt.*;
import java.awt.event.*;
import java.io.*;

public class Pract_16 {
public static void main(String[] args) {
// Create a Frame (window)
Frame frame = new Frame("Layout and Streams Demo");
// Create a layout manager (FlowLayout)
LayoutManager layoutManager = new FlowLayout();
frame.setLayout(layoutManager);
// Create GUI components
Label label = new Label("Enter your name:");
TextField textField = new TextField(20);
Button button = new Button("Submit");
// Add components to the Frame
frame.add(label);
frame.add(textField);
frame.add(button);

// Create an ActionListener for the Button
button.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {

// Get the entered name
String name = textField.getText();
// Write the name to a text file using byte streams
try (FileOutputStream fos = new FileOutputStream("name.txt")) {
byte[] nameBytes = name.getBytes();
fos.write(nameBytes);

System.out.println("Name written to name.txt using byte streams.");

} catch (IOException ex) {
ex.printStackTrace();
}

// Read the name from the text file using character streams

try (FileReader fr = new FileReader("name.txt");
BufferedReader br = new BufferedReader(fr)) {
String readName = br.readLine();
System.out.println("Name read from name.txt using character streams: " +
readName);
} catch (IOException ex) {
ex.printStackTrace();
}
}
});



// Set the Frame size and make it visible
frame.setSize(300, 100);
frame.setVisible(true);



// Add a WindowListener to handle window closing
```

```
frame.addWindowListener(new WindowAdapter() {
public void windowClosing(WindowEvent e) {
 System.exit(0);
}
});
}
}
```



```
[Running] cd "e:\Codes\Practical\JavaCodes\" && javac Pract_16.java && java Pract_16
Name written to name.txt using byte streams.
Name read from name.txt using character streams: Ujjwal Singh
```

Enter your name: Ujjwal Singh