

Program:8 Develop a JAVA program to create an outer class with a function display. Create another class inside the outer class named inner with a function called display and call the two functions in the main class.

Aim: Demonstrate the inheritance along with interface in java programming.

```
public class OuterInner {
    public void display() {
        System.out.println("Outer class display method:");
        System.out.println("In Java, first outer class will
execute... ");
    }

    class Inner {
        public void display() {
            System.out.println("Inner class display method:");
            System.out.println("After outer class, inner class
will execute... ");
        }
    }

    public static void main(String[] args) {
        OuterInner outer = new OuterInner();
        outer.display();
        OuterInner.Inner inner = outer.new Inner();
        inner.display();
    }
}
```

OUTPUT:

```
Outer class display method:
In Java, first outer class will execute...
Inner class display method:
After outer class, inner class will execute...
```

Program:9 Develop a JAVA program to raise a custom exception (user defined exception) for DivisionByZero using try, catch, throw and finally

Aim: Demonstrate creation of user defined exception

```
import java.util.Scanner;

public class Custom_Exception {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter two numbers: ");

        int a = scanner.nextInt();

        int b = scanner.nextInt();

        try {

            if (b == 0) {

                throw new ArithmeticException("Division by zero is not allowed!");

            }

            float result = a / b;

            System.out.println("Result: " + result);

        } catch (ArithmeticException e) {

            System.out.println(e.getMessage());

        } finally {

            scanner.close();

        }

    }

}
```

OUTPUT1:

Enter two numbers:

32

4

Result: 8.0

OUTPUT 2:

Enter two numbers:

4

0

Division by zero is not allowed!

Program:10. Develop a JAVA program to create a package named mypack and import & implement it in a suitable class.

Aim: Demonstrate the IO Package using java programming.

```
// Define the interface Mypackage_interface
```

```
public interface Mypackage_interface {  
    void display(); // Declare a method display()  
}
```

```
// Implement the Mypackage_interface in MyClass
```

```
public class MyClass implements Mypackage_interface {  
    // Implement the display() method  
    public void display() {  
        System.out.println("Hello!");  
        System.out.println("Hi...");  
        System.out.println("You -");  
        System.out.println("Yes You..");  
        System.out.println("Hello, You only.");  
    }  
}
```

```

        System.out.println("this is the result-)-)-");
    }
}

// The main class to demonstrate the usage of MyClass
import mypack.MyClass; // Import MyClass from the package mypack

public class Mypack_main {

    public static void main(String[] args) {

        // Create an object of MyClass

        MyClass obj = new MyClass();

        // Call the display() method of MyClass

        obj.display();

    }

}

```

Output:

```

Hello!
Hi...
You -)
Yes You..
Hello, You only.
this is the result-)-)-

```

Program:11. Write a program to illustrate creation of threads using runnable class. (start method start each of the newly created thread. Inside the run method there is sleep() for suspend the thread for 500 milliseconds).

Aim: Exception handling in java, introduction to throwable class, throw, finally.

```

public class Thread_Example implements Runnable {

```

```

private String name;

public Thread_Example(String name) {
    this.name = name;
}

@Override
public void run() {
    System.out.println("Thread started: " + name);
    try {
        Thread.sleep(500);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    System.out.println("Thread ended: " + name);
}

public static void main(String[] args) {
    Thread_Example runnableExample1 = new Thread_Example("Thread-1");
    Thread_Example runnableExample2 = new Thread_Example("Thread-2");
    Thread_Example runnableExample3 = new Thread_Example("Thread-3");

    Thread thread1 = new Thread(runnableExample1);
    Thread thread2 = new Thread(runnableExample2);
    Thread thread3 = new Thread(runnableExample3);

    thread1.start();
    thread2.start();
    thread3.start();
}
}

```

OUTPUT :

Thread started: Thread-3

Thread started: Thread-2

Thread started: Thread-1

Thread ended: Thread-2

Thread ended: Thread-3

Thread ended: Thread-1

Program:12. Develop a program to create a class MyThread in this class a constructor, call the base class constructor, using super and start the thread. The run method of the class starts after this. It can be observed that both main thread and created child thread are executed concurrently.

Aim: Demonstrate the file operations in java programming.

```
public class MyThread extends Thread {
    public MyThread() {
        super();
        System.out.println("Child Thread");
    }

    @Override
    public void run() {
        try {
            for (int i = 0; i < 5; i++) {
                System.out.println("Child Thread: " + i);
                Thread.sleep(500);
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        MyThread myThread = new MyThread();
        myThread.start();
        try {
            for (int i = 0; i < 5; i++) {
                System.out.println("Main Thread: " + i);
                Thread.sleep(500);
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
```

Output:

```
Child Thread
Child Thread: 0
Main Thread: 0
Child Thread: 1
Main Thread: 1
Child Thread: 2
Main Thread: 2
Child Thread: 3
Main Thread: 3
Child Thread: 4
Main Thread: 4
```