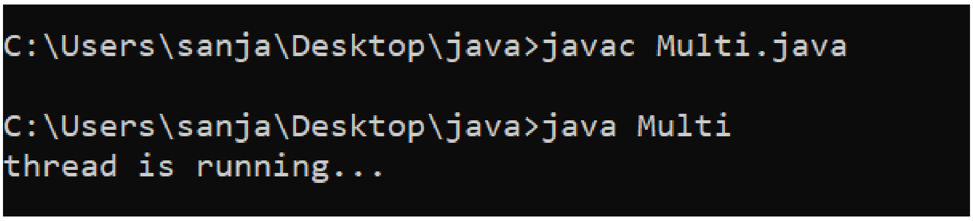
## Practical no. 8:

**Q.1 Write a java program to create Threads :**

**Source Code:**

```
class Multi extends Thread{
public void run(){
System.out.println("thread is running...");
}
public static void main(String args[]){
Multi t1=new Multi();
t1.start();
 }
}
```

**Output:**



```
C:\Users\sanja\Desktop\java>javac Multi.java

C:\Users\sanja\Desktop\java>java Multi
thread is running...
```

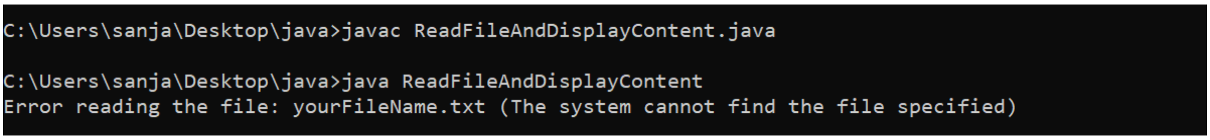
**Q.2 Write a java Program to open a file and display the Content in console window.**

**Source Code :**

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
```

```
public class ReadFileAndDisplayContent {
    public static void main(String[] args) {
        String fileName = "yourFileName.txt";
        try {
            FileReader fileReader = new FileReader(fileName);
            BufferedReader bufferedReader = new BufferedReader(fileReader);
            String line;
            while ((line = bufferedReader.readLine()) != null) {
                System.out.println(line);
            }
            bufferedReader.close();
        } catch (IOException e) {
            System.err.println("Error reading the file: " + e.getMessage());
        }
    }
}
```

**Output:**



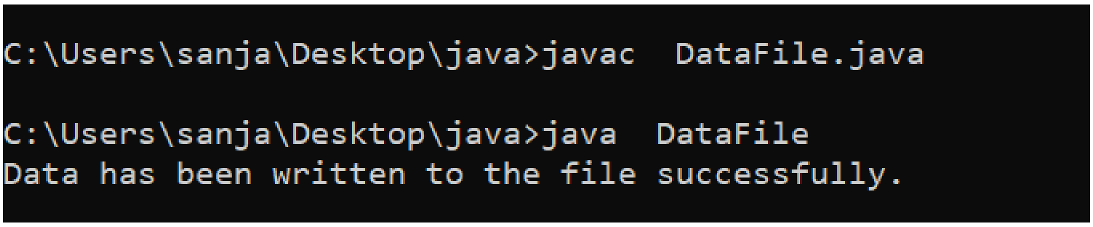
```
C:\Users\sanja\Desktop\java>javac ReadFileAndDisplayContent.java

C:\Users\sanja\Desktop\java>java ReadFileAndDisplayContent
Error reading the file: yourFileName.txt (The system cannot find the file specified)
```

**Q.3 Write a java program to write the data into a file:**

**Source Code :**

```
import java.io.BufferedWriter;

import java.io.FileWriter;

import java.io.IOException;


public class DataFile {

   public static void main(String[] args) {

      String fileName = "output.txt";

      try {

         FileWriter fileWriter = new FileWriter(fileName);

         BufferedWriter bufferedWriter = new
BufferedWriter(fileWriter);

         String data = "Hello, world!\nThis is a Java program to write
data to a file.";

         bufferedWriter.write(data);

         bufferedWriter.close();

         System.out.println("Data has been written to the file
successfully.");

      } catch (IOException e) {

         System.err.println("Error writing to the file: " +
e.getMessage());

      }

   }

}
```

**Output:**



```
C:\Users\sanja\Desktop\java>javac  DataFile.java

C:\Users\sanja\Desktop\java>java  DataFile
Data has been written to the file successfully.
```

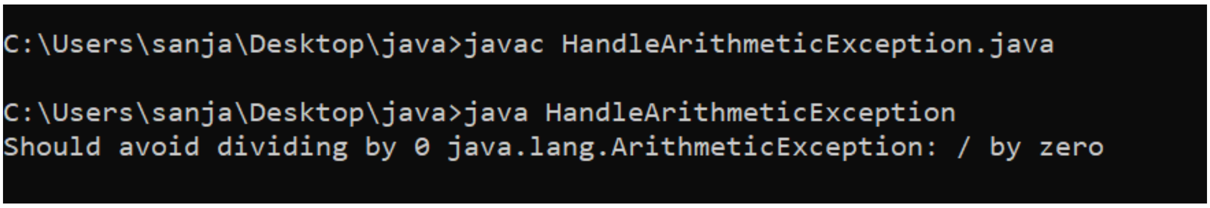
**Practical no.9:**

**Q. Write a java program for performing exception Handling (Arithmetic Exception)**

**Source Code :**

```
public class HandleArithmeticException {
    void divide(int a, int b) {
        int res;
        try {
            // performing division and storing the result
            res = a / b;
            System.out.println("Division process has been done successfully.");
            System.out.println("Result came after division is: " + res);
        } catch (ArithmeticException ex) {
            System.out.println("Should avoid dividing by 0 " + ex);
        }
    }
    public static void main(String[] args) {
        HandleArithmeticException obj = new HandleArithmeticException();
        obj.divide(1, 0);
    }
}
```

**Output:**



```
C:\Users\sanja\Desktop\java>javac HandleArithmeticException.java

C:\Users\sanja\Desktop\java>java HandleArithmeticException
Should avoid dividing by 0 java.lang.ArithmeticException: / by zero
```