**JAVA File**

Java FileInputStream class obtains input bytes from a [file](https://www.javatpoint.com/java-file-class). It is used for reading byte-oriented data (streams of raw bytes) such as image data, audio, video etc. You can also read character-stream data. But, for reading streams of characters, it is recommended to use [FileReader](https://www.javatpoint.com/java-filereader-class) class.

Java FileOutputStream is an output stream used for writing data to a [file](https://www.javatpoint.com/java-file-class). If you have to write primitive values into a file, use FileOutputStream class. You can write byte-oriented as well as character-oriented data through FileOutputStream class. But, for character-oriented data, it is preferred to use [FileWriter](https://www.javatpoint.com/java-filterwriter-class) than FileOutputStream.

Java FileReader class is used to read data from the file. It returns data in byte format like [FileInputStream](https://www.javatpoint.com/java-fileinputstream-class) class. It is character-oriented class which is used for [file](https://www.javatpoint.com/java-file-class) handling in [java](https://www.javatpoint.com/java-tutorial).

Java FileWriter class is used to write character-oriented data to a [file](https://www.javatpoint.com/java-file-class). It is character-oriented class which is used for file handling in [java](https://www.javatpoint.com/java-tutorial). Unlike FileOutputStream class, you don't need to convert string into byte [array](https://www.javatpoint.com/array-in-java) because it provides method to write string directly

**Program 1**

**import** java.io.\*;

**public** **class** FileDemo {

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

**try** {

            File file = **new** File("javaFile123.txt");

**if** (file.createNewFile()) {

                System.out.println("New File is created!");

            } **else** {

                System.out.println("File already exists.");

            }

        } **catch** (IOException e) {

            e.printStackTrace();

        }

    }

}

**Program 2**

**import java.io.\*;**

**public class FileDemo2 {**

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

**String path = "";**

**boolean bool = false;**

**try {**

**// createing  new files**

**File file  = new File("testFile1.txt");**

**file.createNewFile();**

**System.out.println(file);**

**// createing new canonical from file object**

**File file2 = file.getCanonicalFile();**

**// returns true if the file exists**

**System.out.println(file2);**

**bool = file2.exists();**

**// returns absolute pathname**

**path = file2.getAbsolutePath();**

**System.out.println(bool);**

**// if file exists**

**if (bool) {**

**// prints**

**System.out.print(path + " Exists? " + bool);**

**}**

**} catch (Exception e) {**

**// if any error occurs**

**e.printStackTrace();**

**}**

**}**

**}**

**Program 3**

**import java.io.\*;**

**class CreateFile**

**{**

**public static void main(String[] args) throws IOException**

**{**

**// Accept a string**

**String str = "File Handling in Java using "+**

**" FileWriter and FileReader";**

**// attach a file to FileWriter**

**FileWriter fw=new FileWriter("output.txt");**

**// read character wise from string and write**

**// into FileWriter**

**for (int i = 0; i < str.length(); i++)**

**fw.write(str.charAt(i));**

**System.out.println("Writing successful");**

**//close the file**

**fw.close();**

**}**

**}**

**Program 4**

**import java.io.\*;**

**class ReadFile**

**{**

**public static void main(String[] args) throws IOException**

**{**

**// variable declaration**

**int ch;**

**// check if File exists or not**

**FileReader fr=null;**

**try**

**{**

**fr = new FileReader("text");**

**}**

**catch (FileNotFoundException fe)**

**{**

**System.out.println("File not found");**

**}**

**// read from FileReader till the end of file**

**while ((ch=fr.read())!=-1)**

**System.out.print((char)ch);**

**// close the file**

**fr.close();**

**}**

**}**

**Program 5**

**package** com.javatpoint;

**import** java.io.FileInputStream;

**public** **class** DataStreamExample {

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

**try**{

            FileInputStream fin=**new** FileInputStream("D:\\testout.txt");

**int** i=0;

**while**((i=fin.read())!=-1){

             System.out.print((**char**)i);

            }

            fin.close();

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

         }

        }

**Program 6**

**import** java.io.FileOutputStream;

**public** **class** FileOutputStreamExample {

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

**try**{

             FileOutputStream fout=**new** FileOutputStream("D:\\testout.txt");

             String s="file output stream an output stream used for writing data to a [file](https://www.javatpoint.com/java-file-class)";

**byte** b[]=s.getBytes();//converting string into byte array

             fout.write(b);

             fout.close();

             System.out.println("success...");

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

      }

}

**Program 7:**

**Write a JAVA program to create an interface Stack and create a class StackOp that implements the interface Stack to perform push and pop operations.**

**Do it by yourself**