© MSSQUARE Global Corporate Training Institute Contact: 8087883669

File

https://docs.oracle.com/javase/8/docs/api/java/io/File.html

Note: must read java doc apis.

```
package com.hdfc.file;

import java.io.*;
import java.util.Scanner;


/**
 *
 *   /  looks like F Forward slash
 *  \ looks like B Backword slash
 *  > look like g greate than
 *  < look like L less than
 *
 *  windows :  backword slash \\
 *  unix/linux: forward slash /
 */


//https://docs.oracle.com/javase/8/docs/api/java/io/File.html
public class FileTest {

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

        //parent pathname string and a child pathname string.
        //"C:\\Users\\sutha\\Desktop\\test" is directory
        //"Ravi2.txt" is file name

        //or
        // C:\Users\sutha\Desktop parent path
        // \test\Ravi2.txt is child path

        //File file = new File("C:\\Users\\sutha", "\\Desktop\\test\\Ravi2.txt");
        //System.out.println(file.createNewFile());
        //file.mkdir();
        //System.out.println(file.exists());//true?


        //File reprent a path for a file name
        //File file = new File("C:\\Users\\sutha\\Desktop\\test\\MyFolder110\\MyFolder120\\MyFolder130");
       // System.out.println(file.exists());
        //System.out.println(file.mkdirs()); //mkdir for create folder
        //mkdirs to create nexted folders



        //unix /Linux
        //read write execute
        /*file.canRead(); //to check if file is readable or not?
        file.canWrite();//to check if file is writable or not?
        file.canExecute();//to check if file is executable or not?

        if(true){//admin
            file.setReadable(true);
            file.setWritable(true);
            file.setExecutable(true);
        }else{
            file.setReadable(true);
            file.setWritable(false); //not writable
```

© MSSQUARE Global Corporate Training Institute Contact: 8087883669

```
            file.setExecutable(false); //.sh non executable
        }*/

        //An abstract representation of file and directory pathnames.
       // File file = new File("TestJava.txt"); //a file name
       // File file2 = new File("C:\\Users\\sutha");//path
       // file.createNewFile();
      //  System.out.println(file.getAbsolutePath());
      //  System.out.println(file.getCanonicalPath());


      //  String hello ="  \\Mayank\\  "; // "  \"Mayank\"  "
      //  System.out.println(hello);



        // Stream based-Stream (Input-read, Output-write)
        //Character based (text based)- Reader/Writer (Input-read, Output-write)

        //PDF pdf lib
        //Excel apache poi


        /*File  dir = new File("C:\\Users\\sutha\\Desktop\\test\\UserFolder\\Temp");
        System.out.println(dir.exists());//false
        if(!dir.exists()){
            dir.mkdirs();//nested folder will be created.
        }
        System.out.println(dir.exists());//true*/


     /*   File dir = new File("C:\\Users\\sutha\\Desktop\\test");
        String[] list = dir.list();
        for(String fileOrDirectory: list){
            System.out.println(fileOrDirectory);
        }
        System.out.println(list.length);//all files in folder
*/

        File dir = new File("C:\\Users\\sutha\\Desktop\\test");
        String[] list = dir.list(new FilenameFilter() {
            @Override
            public boolean accept(File dir, String name) {
                return name.contains(".class");//to get files having .class extension
            }
        });
        for(String fileOrDirectory: list){
            System.err.println(fileOrDirectory);
        }
        System.err.println(list.length);//all files count in folder



    }
}
```

© MSSQUARE Global Corporate Training Institute Contact: 8087883669

PriorityQueue – poll in reverse number order, (highest number first poll)

```
package com.hdfc.operator;


//Queue - FIFO
//first in first out

import com.hdfc.collections.User;

import java.util.Comparator;
import java.util.Date;
import java.util.List;
import java.util.PriorityQueue;

class UserIdComparator implements Comparator<User> {

    @Override
    public int compare(User o1, User o2) {
        if (o1.getId() == o2.getId()) {
            return 0;
        } else if (o1.getId() < o2.getId()) {
            return 1;
        } else {
            return +1;
        }
    }
}

//5 2 1 3 6
//1 2 3 4 5
public class PriorityQueueTest {
    public static void main(String[] args) {

        //User class should implements Comparable interface
        // If User class is from library then? Comparator interface
        PriorityQueue<User> priorityQueue = new PriorityQueue<>(new UserIdComparator());
        priorityQueue.add(new User(5, "F", List.of("F"), new Date()));
        priorityQueue.add(new User(3, "C", List.of("C"), new Date()));
        priorityQueue.add(new User(1, "A", List.of("A"), new Date()));
        priorityQueue.add(new User(2, "B", List.of("B"), new Date()));

        System.out.println(priorityQueue.size());
        System.out.println(priorityQueue.poll()); //poll get and remove
        System.out.println(priorityQueue.size());

        System.out.println(priorityQueue.peek()); //just check the first elemetn in queue
        System.out.println(priorityQueue.size());

      /*  PriorityQueue<Integer> priorityQueue = new PriorityQueue<>(new NumberCompartor());
        priorityQueue.add(5);
        priorityQueue.add(2);
        priorityQueue.add(1);
        priorityQueue.add(3);
        priorityQueue.add(6);

        System.out.println(priorityQueue.poll());
        System.out.println(priorityQueue.poll());
        System.out.println(priorityQueue.poll());
        System.out.println(priorityQueue.poll());
        System.out.println(priorityQueue.poll());*/
```

© MSSQUARE Global Corporate Training Institute Contact: 8087883669

```
    }
}

class NumberCompartor implements Comparator<Integer>{
    @Override
    public int compare(Integer o1, Integer o2) {
        return - o1.compareTo(o2);
    }
}
```

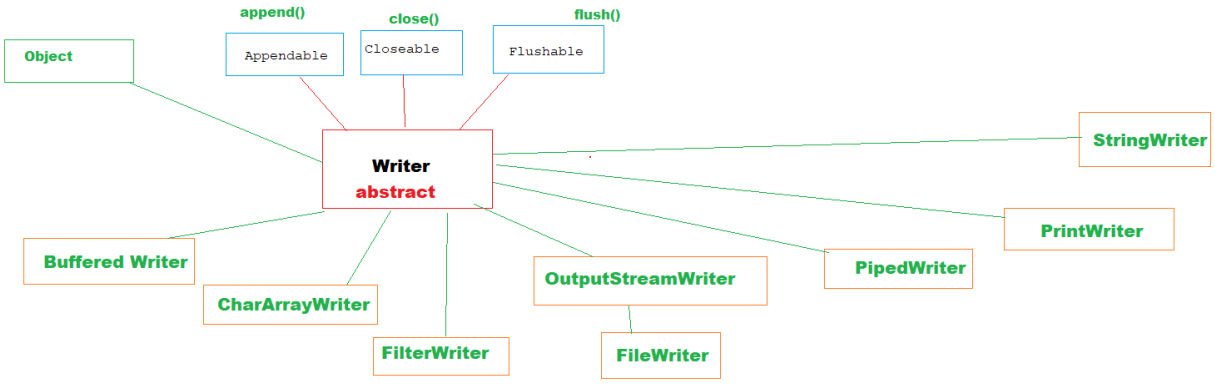
FileWriter and FileReader

```
package com.hdfc.file;

import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class TestFile {


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


       /* File file = new File("abc.txt");
        FileWriter fw = new FileWriter(file); // new file will be created. override
        FileWriter fw2= new FileWriter("abc.txt");// new file will be created. override

        FileWriter fw3 = new FileWriter("abc.txt",true); // append to existing file
        FileWriter fw4= new FileWriter(file, true);// append to existing file


        fw4.write("Nikit");*/



        File file = new File("abc.txt");
        FileWriter fw = new FileWriter(file);
        fw.write("100");
        fw.write('\n'); //new line
        fw.write(100);//https://www.cs.cmu.edu/~pattis/15-1XX/common/handouts/ascii.html
        fw.write('\n'); //new line
        fw.write('c');
        fw.flush();
        fw.close();

        FileReader fr = new FileReader(file);
        int character ;
        while( (character= fr.read()) !=-1) {
            System.out.print((char)character);
        }
        fr.close();


    }
}
```

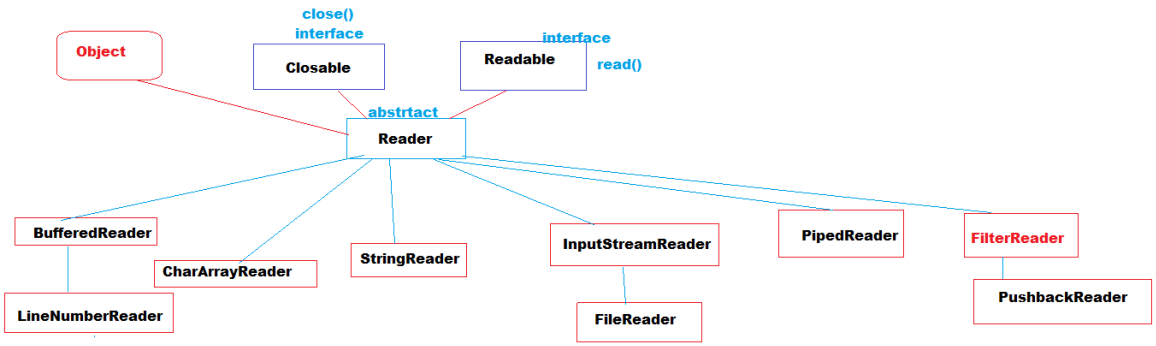
© MSSQUARE Global Corporate Training Institute Contact: 8087883669

- File IO, IO Stream, java.i.o, java io all are same,
- java.io package available from java 1.0 version
- From java 1.4 new package java.nio is introduced
- java.nio is not a replacement of java.io package, both performing in different ways
- java.io package worked on Stream based
- java.nio on buffered based

- Java.io has 2 categories
- Part -1 for character based, text based, we used FileReader, FileWriter, BufferedReader, Bufered Writer, etc
- Part -2 for Byte Stream, for video, audio, images, OutputStream inputStream related classes
- Output is to write data on sink(file, destionation)
- Input is for read the data from file, source system.



© MSSQUARE Global Corporate Training Institute Contact: 8087883669

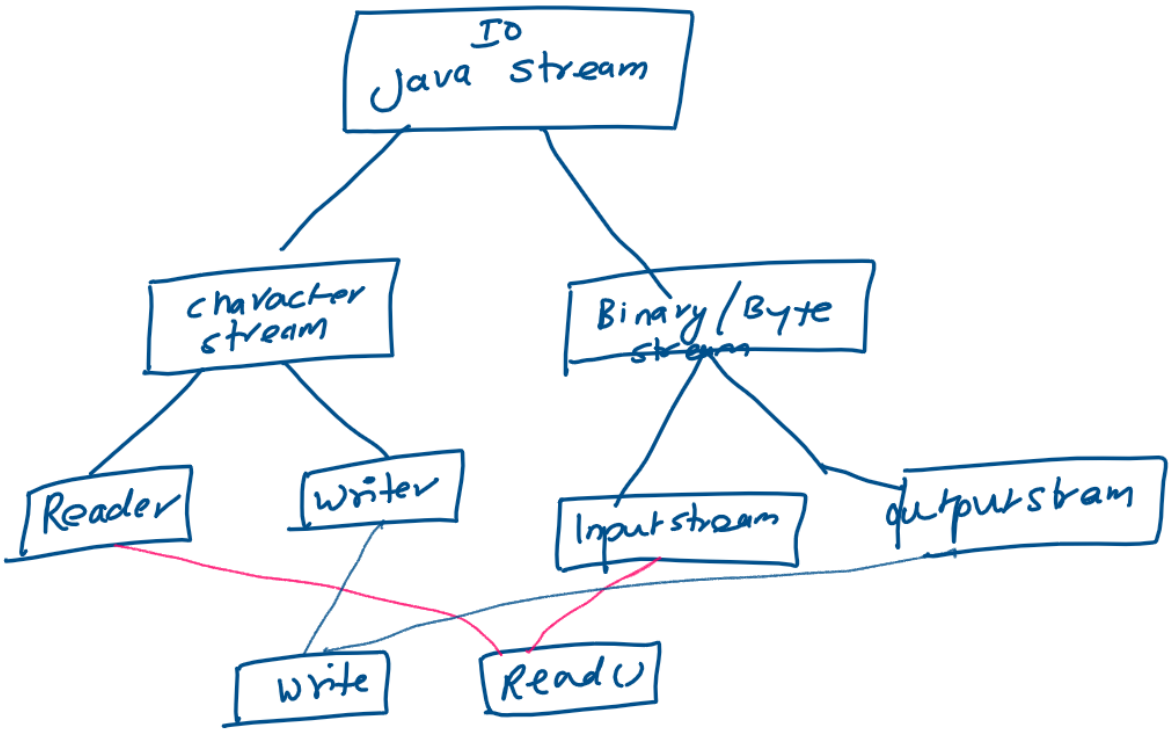


FileReader and FileWriter work on char to char basis.

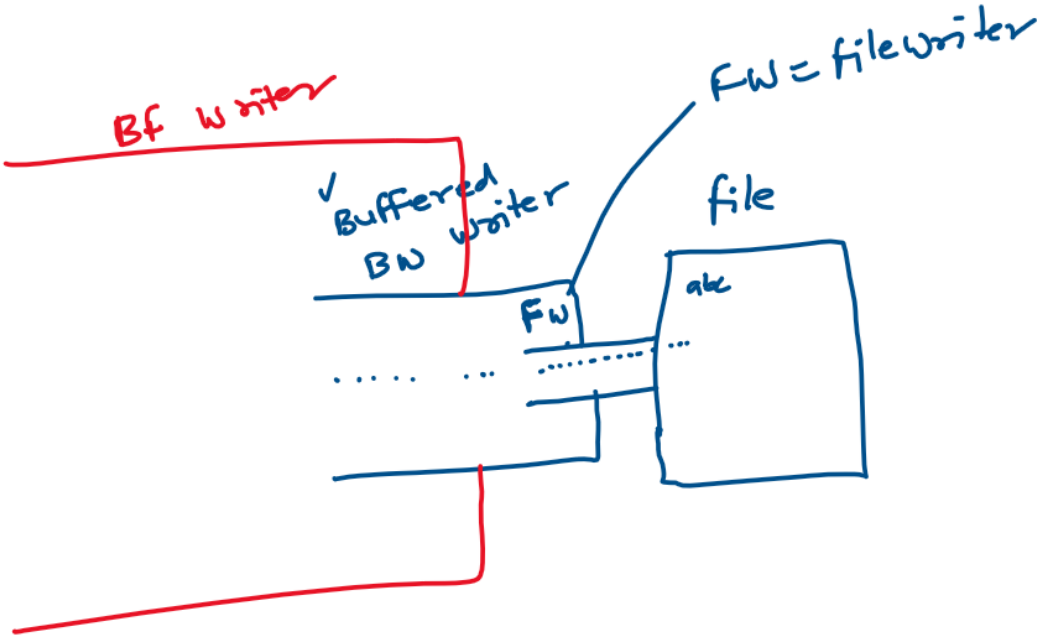
```
package com.hdfc.file;

import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class TestFile {


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


       /* File file = new File("abc.txt");
        FileWriter fw = new FileWriter(file); // new file will be created. override
        FileWriter fw2= new FileWriter("abc.txt");// new file will be created. override

        FileWriter fw3 = new FileWriter("abc.txt",true); // append to existing file
        FileWriter fw4= new FileWriter(file, true);// append to existing file


        fw4.write("Nikit");*/



        File file = new File("abc.txt");
        FileWriter fw = new FileWriter(file);
        fw.write("100");
        fw.write('\n'); //new line
        fw.write(100);//https://www.cs.cmu.edu/~pattis/15-1XX/common/handouts/ascii.html
        fw.write('\n'); //new line
        fw.write('c');
        fw.flush();
        fw.close();

        FileReader fr = new FileReader(file);
        int character ;
        while( (character= fr.read()) !=-1) {
            System.out.print((char)character);
        }
        fr.close();
```

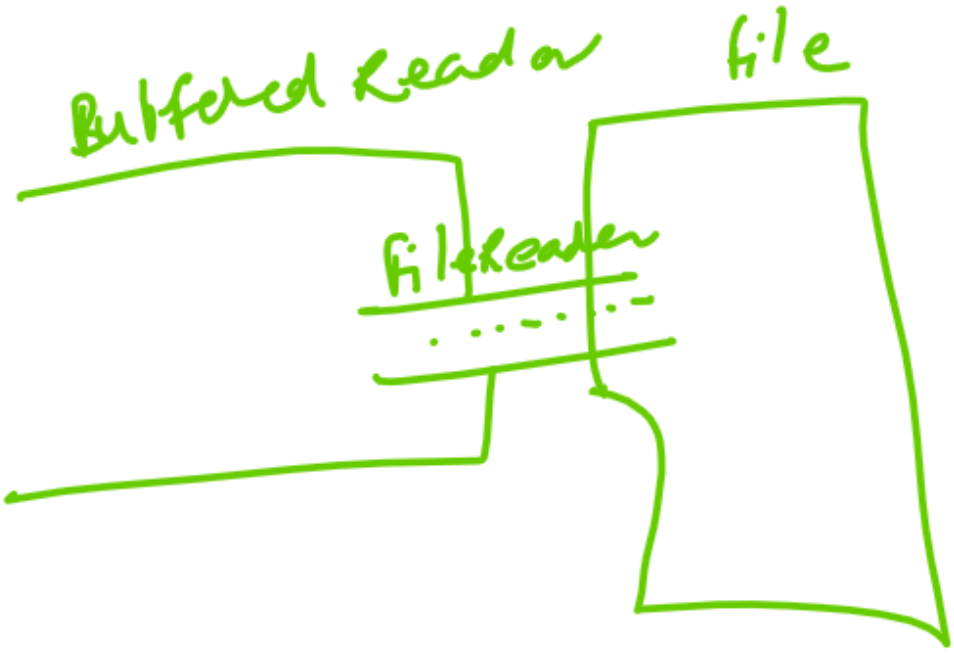
© MSSQUARE Global Corporate Training Institute Contact: 8087883669

```
    }
}
```

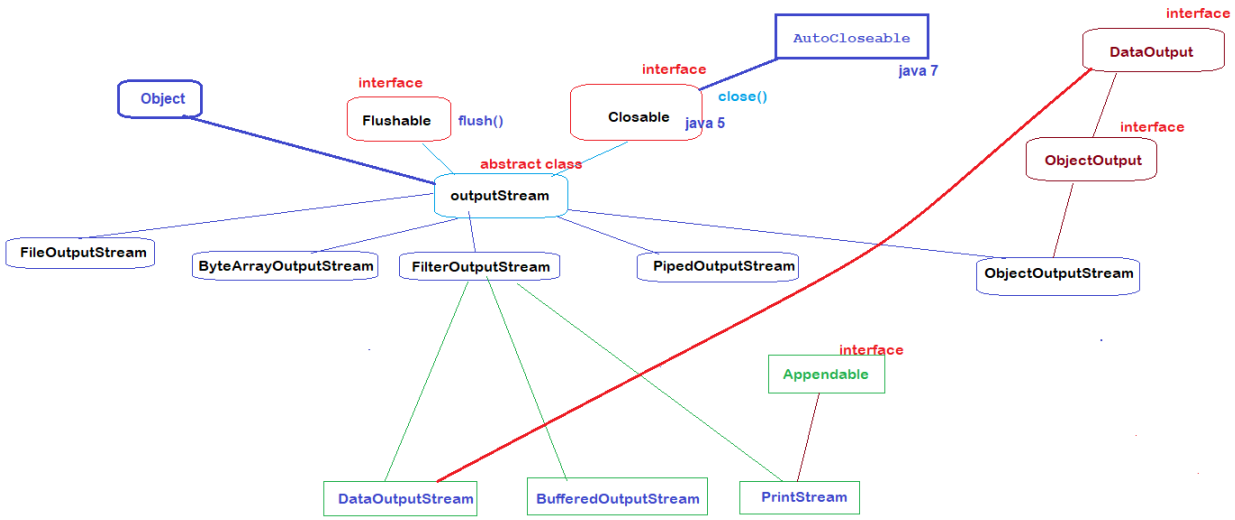


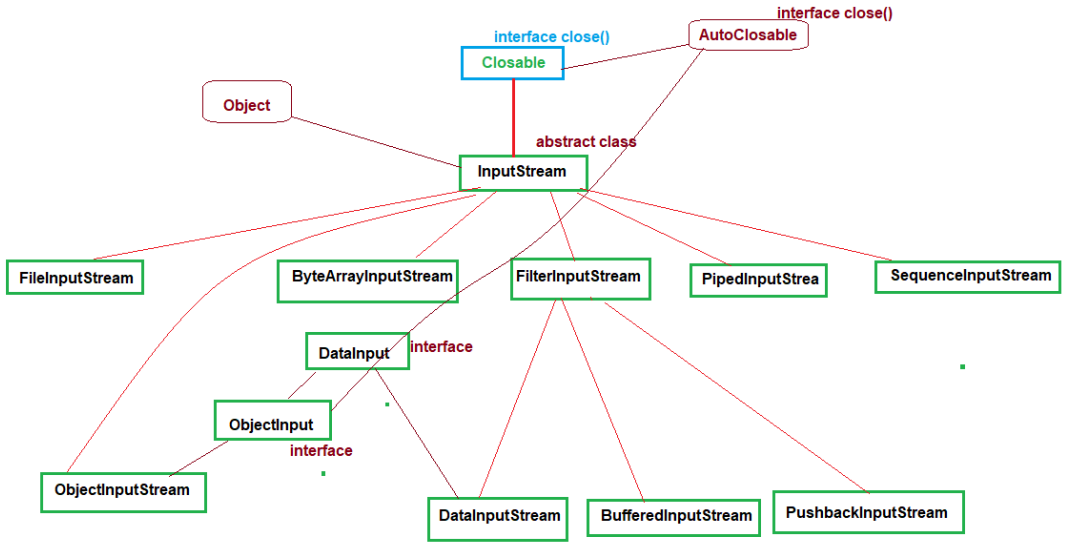
© MSSQUARE Global Corporate Training Institute Contact: 8087883669





© MSSQUARE Global Corporate Training Institute Contact: 8087883669





© MSSQUARE Global Corporate Training Institute Contact: 8087883669

```java
package com.hdfc.file;

import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class TestFile {


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


       /* File file = new File("abc.txt");
        FileWriter fw = new FileWriter(file); // new file will be created. override
        FileWriter fw2= new FileWriter("abc.txt");// new file will be created. override

        FileWriter fw3 = new FileWriter("abc.txt",true); // append to existing file
        FileWriter fw4= new FileWriter(file, true);// append to existing file
        fw4.write("Nikit");*/

        File file = new File("abc.txt");
        FileWriter fw = new FileWriter(file);
        fw.write("100");
        //fw.write() 100 as a number, no method.
        //float 10.10f write. no method.
        fw.write('\n'); //new line   \n
        fw.write(100);//https://www.cs.cmu.edu/~pattis/15-1XX/common/handouts/ascii.html
        fw.write('\n'); //new line
        fw.write('c');
        fw.write('\n'); //new line
        fw.write(':');//https://www.cs.cmu.edu/~pattis/15-1XX/common/handouts/ascii.html
        fw.flush();
        fw.close();

        //65 - 90 A-Z
        //97-122 a-z

        FileReader fr = new FileReader(file);
       /*int character ;
        while( (character= fr.read()) !=-1) {
            System.out.print((char)character);
        }*/



       /* int character=fr.read();
        while(character!=-1){
            System.out.print((char)character);
            character=fr.read();
        }
```

© MSSQUARE Global Corporate Training Institute Contact: 8087883669

```
        fr.close();*/


        char[] charArray = new char[(int)file.length()]; //file has 4 chars
        int read = fr.read(charArray);
        for(char c: charArray){
            System.out.println(c);
        }

    }
}
```

```
package com.hdfc.file;

import java.io.*;

public class BufferedTest {

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

        File file = new File("buffered.txt");
        FileWriter fw = new FileWriter(file);
        BufferedWriter bw = new BufferedWriter(fw);
        //BufferedWriter bf = new BufferedWriter(new BufferedWriter(fw)); just to show you

        bw.write("This is my first line of sentence.");
        bw.newLine();  //'\n'
        bw.write(200);
        bw.newLine();

        //100 -> c
        // 100 -> "100" BufferedWriter does not have method to write only interger numbers
        bw.write("100");
        //10.10 -> "10.10"   BufferedWriter does not have method to write only decimal numbers
        bw.write("10.10");
        //true  -> "true" BufferedWriter does not have method to write only boolean
        bw.write("true");

        bw.flush();
        bw.close();

        /**
         * case 1 bw.close()  -- this will also close fileWrite
         * case 2 fw.close()
         * case 3 fw.clos() and then bw.close()
         *
         */



        File file2 = new File("buffered.txt");
        FileReader fr = new FileReader(file2);
        BufferedReader br = new BufferedReader(fr);

        String s = br.readLine(); //get the data in line
        while(s != null){
            System.out.print(s);
            s = br.readLine();
        }

        br.close();
    }
}
```

© MSSQUARE Global Corporate Training Institute Contact: 8087883669

```java
package com.hdfc.file;

import java.io.*;

public class PrintWriterTest {

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

        FileWriter fw = new FileWriter("print.txt");
        PrintWriter pw = new PrintWriter(fw);

        pw.write("write method");

        //use print or println methods. not write method from Writer class
        pw.println("hello");
        pw.println(100);
        pw.println(10.10F);
        pw.println(true);

        pw.flush();
        pw.close();


        //there is no PrintReader
        //PrintReader


        //Writer
        //FileWriter  -write char by char, for new link \n
        //BufferedWriter - newLine() method for \n
        //PrintWriter  - this is good to use , have methods for print all primitives and println
method for line

        //Read
        //FileReader - read char by car
        //BufferedReader - this is good to use , read line by line
        //we don't have printReader class.

        File file2 = new File("print.txt");
        FileReader fr = new FileReader(file2);
        BufferedReader br = new BufferedReader(fr);

        String s = br.readLine();
        while(s != null){
            System.out.println(s);
            s = br.readLine();
        }

        br.close();
    }
}
```

```java
package com.hdfc.file;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;

public class OutputStreamTest {

    public static void main(String[] args) {


        //to write
```

© MSSQUARE Global Corporate Training Institute Contact: 8087883669

```
        File file = new File("outputstream.txt");
        //java 7 - try with resource introduced in java 7
        //try () only autoClosable we can write.
        try (FileOutputStream fos = new FileOutputStream(file)) {

            fos.write(100);//d
            fos.write('\n');
            fos.write('c'); //c

            fos.flush();
        } catch (Exception e) {
            System.err.println(e);
        }


        //folder1 /img.png
        //folder2 /img.png


        //to read
        //java 7 - try with resource introduced in java 7
        try (FileInputStream fis = new FileInputStream("outputstream.txt")) {
            int character = fis.read();
            while (character != -1) {
                System.out.print((char) character);
                character = fis.read();
            }
        } catch (Exception e) {
            System.err.println(e);
        }
    }


    //java 1.1  jdk 1  java 1
    //java 1.5  jdk 5  java 5
    //java 1.8 jdk 8   java 8
    //java 9  jdk 9
    //java 17 jdk 17
}
```

```
package com.hdfc.file;

import java.io.*;

public class BufferedTest {

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

        File file = new File("buffered.txt");
        FileWriter fw = new FileWriter(file);
        BufferedWriter bw = new BufferedWriter(fw);
        //BufferedWriter bf = new BufferedWriter(new BufferedWriter(fw)); just to show you

        bw.write("This is my first line of sentence.");
        bw.newLine();  //'\n'
        bw.write(200);
        bw.newLine();

        //100 -> c
        // 100 -> "100" BufferedWriter does not have method to write only interger numbers
        bw.write("100");
        //10.10 -> "10.10"   BufferedWriter does not have method to write only decimal numbers
        bw.write("10.10");
```

© MSSQUARE Global Corporate Training Institute Contact: 8087883669

```
        //true  -> "true" BufferedWriter does not have method to write only boolean
        bw.write("true");

        bw.flush();
        bw.close();

        /**
         * case 1 bw.close()  -- this will also close fileWrite
         * case 2 fw.close()
         * case 3 fw.clos() and then bw.close()
         *
         */



       /* File file2 = new File("buffered.txt");
        FileReader fr = new FileReader(file2);
        BufferedReader br = new BufferedReader(fr);

        String s = br.readLine(); //get the data in line
        while(s != null){
            System.out.print(s);
            s = br.readLine();
        }

        br.close();*/


        File file2 = new File("buffered.txt");
        FileReader fr = new FileReader(file2);
        BufferedReader br = new BufferedReader(fr);
        StringWriter sw = new StringWriter();

        String s = br.readLine(); //get the data in line
        while(s != null){
            //System.out.print(s);
            sw.write(s);
            s = br.readLine();
        }

        br.close();

        System.out.println(sw.toString());
    }
}
```

```
package com.hdfc.file;

import java.io.ByteArrayOutputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class CharArrayStreamTest {

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

        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        baos.write(100);// 'd'

        FileOutputStream fos = new FileOutputStream("Boas.txt");
        baos.writeTo(fos);
        fos.flush();
        fos.close();
```

© MSSQUARE Global Corporate Training Institute Contact: 8087883669

```
    }
}
```

```
package com.hdfc.file;

import java.io.*;

public class ObjectOutputStreamTest {

    public static void main(String[] args) throws IOException, ClassNotFoundException {

        String source ="Hello World!";
        //write the live object
        FileOutputStream fos = new FileOutputStream("filename.txt");
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        oos.writeObject(source);
        oos.flush();
        oos.close();

        //read the live object
        FileInputStream fis = new FileInputStream("filename.txt");
        ObjectInputStream ois = new ObjectInputStream(fis);
        String destination = (String)ois.readObject();
        ois.close();
        System.out.println(destination);
    }

}
```

```
package com.hdfc.file;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;

public class OutputStreamTest {

    public static void main(String[] args) {


        //to write
        File file = new File("outputstream.txt");
        //java 7 - try with resource introduced in java 7
        //try () only autoClosable we can write.
        try (FileOutputStream fos = new FileOutputStream(file)) {

            fos.write(100);//d
            fos.write('\n');
            fos.write('c'); //c

            fos.flush();
        } catch (Exception e) {
            System.err.println(e);
        }


        //folder1 /img.png
        //folder2 /img.png


        //to read
        //java 7 - try with resource introduced in java 7
        try (FileInputStream fis = new FileInputStream("outputstream.txt")) {
            int character = fis.read();
```

© MSSQUARE Global Corporate Training Institute Contact: 8087883669

```
            while (character != -1) {
                System.out.print((char) character);
                character = fis.read();
            }
        } catch (Exception e) {
            System.err.println(e);
        }
    }


    //java 1.1  jdk 1  java 1
    //java 1.5  jdk 5  java 5
    //java 1.8 jdk 8   java 8
    //java 9  jdk 9
    //java 17 jdk 17
}
```

```
package com.hdfc.file;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;

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


        //read the write
        //C:\Users\sutha\Desktop\MSSquareGlobal\Reader.png
        System.out.println("reading file to source");
        File file = new File("C:\\Users\\sutha\\Desktop\\MSSquareGlobal\\Reader.png");
        byte[] byteArray = new byte[(int) file.length()];
        try (FileInputStream fis = new FileInputStream(file)) {
            fis.read(byteArray);
        } catch (Exception e) {
            System.err.println();
        }

        //write the file
        //write to different destination
        System.out.println("writing file to destination");
        File destinationPath = new File("MyReaderImage.png");
        try (FileOutputStream fos = new FileOutputStream(destinationPath)) {
            fos.write(byteArray);
            fos.flush();
        } catch (Exception e) {
            System.err.println();
        }

    }
}
```