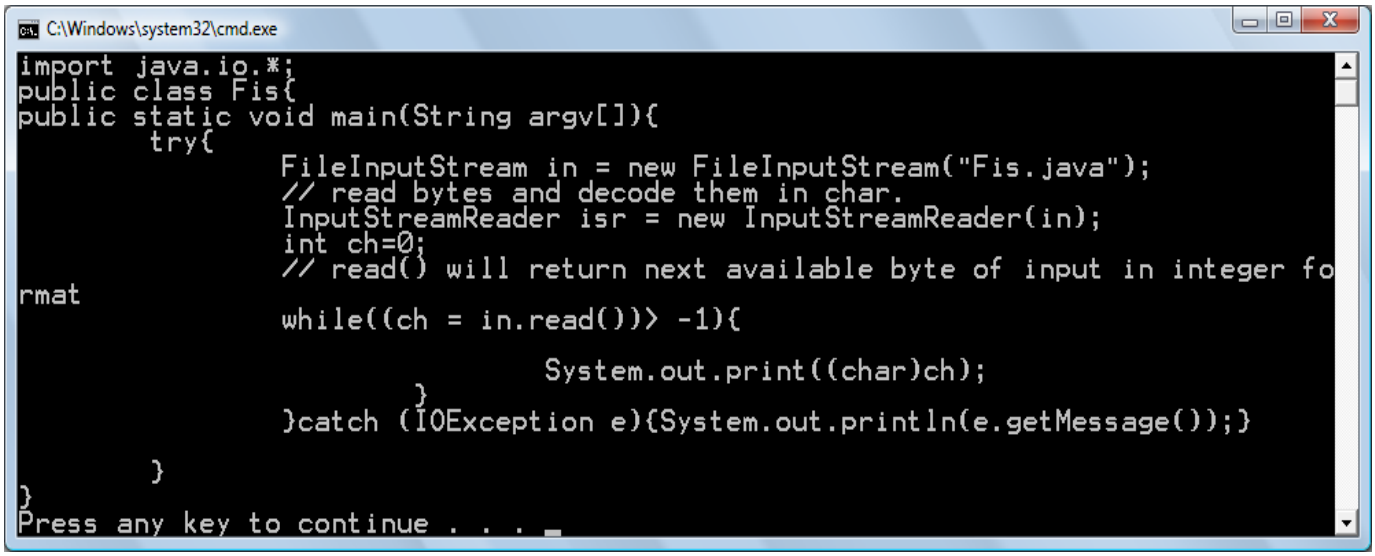
Code Example of Streams (Bytes & Char) and Serialization

Example # 01 (Read File with Byte stream?)

Fis.java

```
import java.io.*;
public class Fis{
public static void main(String argv[]){
        try{
                FileInputStream in = new FileInputStream("Fis.java");
                // read bytes and decode them in char.
        InputStreamReader isr = new InputStreamReader(in);
                int ch=0;
                // read() will return next available byte of input in integer format
                while((ch = in.read())> -1){

                        System.out.print((char)ch);
                        }
                }catch (IOException e){System.out.println(e.getMessage());}
        }
}
```

Output:

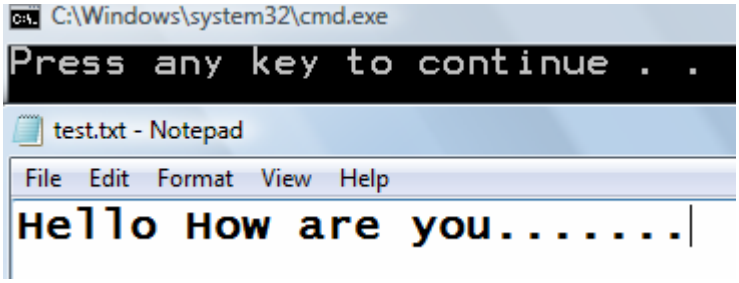


Example # 02 (Write File with Byte stream?)

Fos.java

```
import java.io.*;

public class Fos{

      String s = new String("Hello How are you.......");

          public static void main(String argv[]){
                Fos f = new Fos();
                f.amethod();
                   }

            public void amethod(){
                try{
                FileOutputStream fos = new FileOutputStream("test.txt");
                byte buf[] = s.getBytes();
                for(int i=0; i<buf.length; i++ )
                {
                fos.write(buf[i]);
                }
                }catch(IOException ioe) {}
              }
}
```

Output:

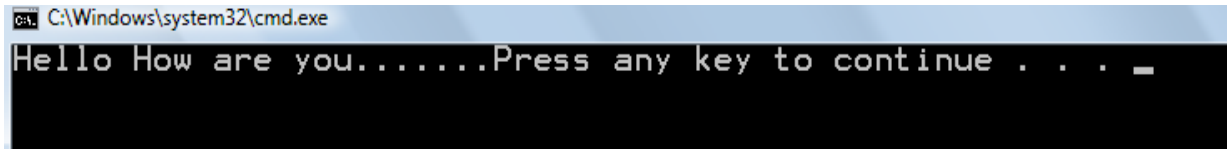


Example # 03 (Read file through "Buffer" with Byte streams)

BufIn.java

```
import java.io.*;
public class BufIn{
public static void main(String argv[]){
        try{
                FileInputStream fin = new FileInputStream("test.txt");
                BufferedInputStream bin = new BufferedInputStream(fin);
                int ch=0;
                while((ch=bin.read())> -1){
                        System.out.print((char)ch);
                        }

                }catch(IOException e){System.out.println(e.getMessage());};
        }
}
```

Output:

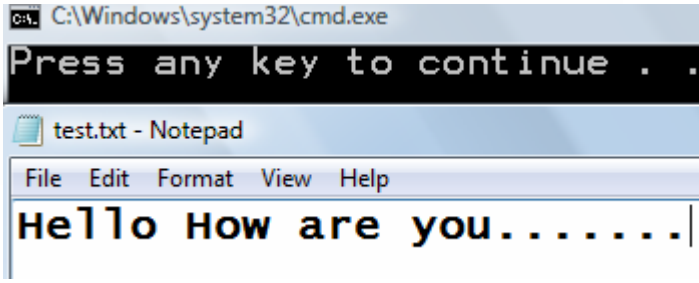


Example # 04 (Write file through "Buffer" with Byte streams)

Bos.java

```
import java.io.*;

public class Bos {

    public void writeToFile(String filename) {

        BufferedOutputStream bufferedOutput = null;

        try {
            //Construct the BufferedOutputStream object
            bufferedOutput = new BufferedOutputStream(new FileOutputStream(filename));

            //Start writing to the output stream
            bufferedOutput.write("Hello How are you.......".getBytes());

            //close BOS so that it can write bytes on FOS
            bufferedOutput.close();

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

    public static void main(String[] args) {
        new Bos().writeToFile("test.txt");
    }
}
```

Output:

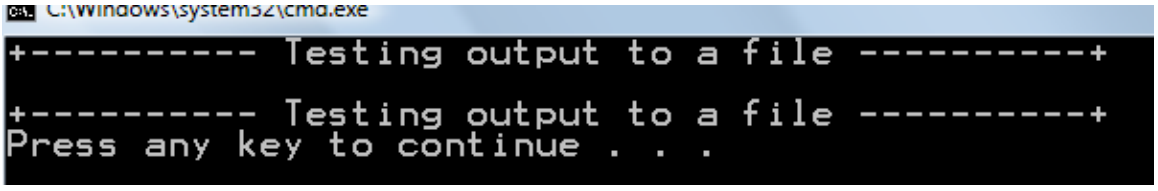


Example # 05 (Read file with Char stream)

ReadinToFile.java

```
import java.io.*;

public class ReadingToFile {

    private static void doRead() {

        try {
            String fileName = "WritingToFile.txt";
            BufferedReader in = new BufferedReader(new FileReader(fileName));
                    /* Reading char by char
                    int c = 0;
                    while ( (c=in.read()) != -1)
                    {
                        System.out.print((char)c);
                    }*/
               String s;
               while((s = in.readLine()) != null) {
               System.out.println(s);
                }
               in.close();

        } catch (IOException e) {

            System.out.println("IOException:");
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        doRead();
    }
}
```

Output:

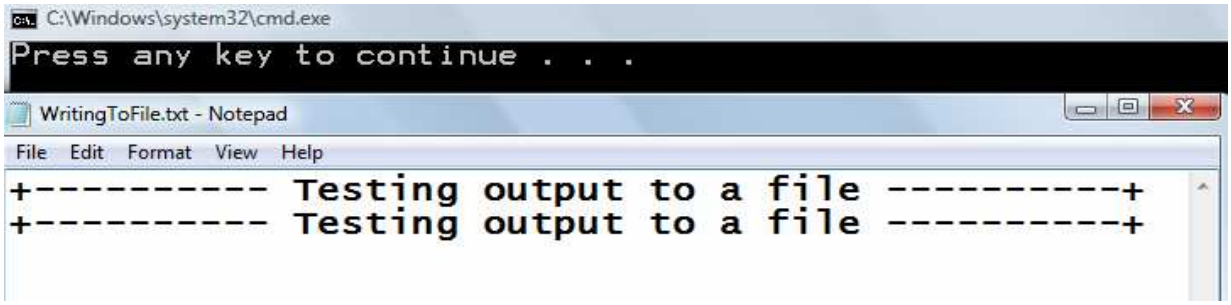


Example # 06 (Write file with Char stream)

WritingToFile.java

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;

public class WritingToFile {

    private static void doWrite() {
        try {
            String fileName = "WritingToFile.txt";
            BufferedWriter out = new BufferedWriter(new FileWriter(fileName));
            out.write("+---------- Testing output to a file ----------+");
           // Print several new line characters. I use two styles here.
            out.write("\n");
            out.write("\n");
            out.write("+---------- Testing output to a file ----------+");
            out.close();
        } catch (IOException e) {
            System.out.println("IOException:");
            e.printStackTrace();
        }

    }

    public static void main(String[] args) {
        doWrite();
    }

}
```

Output:



Example # 06 (Serialization Example)

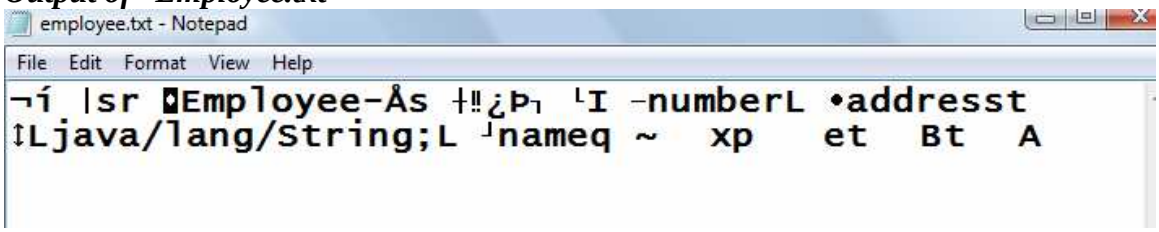
Employee.java

```
import java.io.*;

class Employee implements java.io.Serializable {
  public String name;
  public String address;
  public transient int SSN;

  public int number;

  public void mailCheck() {
    System.out.println("Mailing a check to " + name + " " + address);
  }
}
```

SerializeDemo.java

```
import java.io.*;

 class SerializeDemo {
  public static void main(String[] args) {
    Employee e = new Employee();
    e.name = "A";
    e.address = "B";
    e.SSN = 11111;
    e.number = 101;
    try {
      FileOutputStream fileOut = new FileOutputStream("employee.txt");
      ObjectOutputStream out = new ObjectOutputStream(fileOut);
      out.writeObject(e);
      out.close();
      fileOut.close();
    } catch (IOException i) {
      i.printStackTrace();
    }
  }
}
```

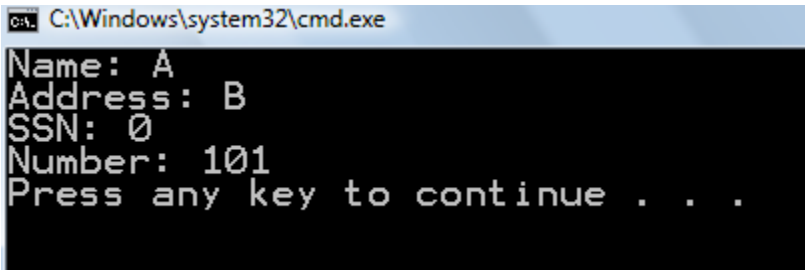
Output of "Employee.txt"



DeserializeDemo.java

```
import java.io.*;

class DeserializeDemo {
  public static void main(String[] args) {
    Employee e = null;
    try {
      FileInputStream fileIn = new FileInputStream("employee.txt");
      ObjectInputStream in = new ObjectInputStream(fileIn);
      e = (Employee) in.readObject();
      in.close();
      fileIn.close();
    } catch (IOException i) {
      i.printStackTrace();
    } catch (ClassNotFoundException c) {
      System.out.println("Employee class not found");
      c.printStackTrace();
      return;
    }
    System.out.println("Name: " + e.name);
    System.out.println("Address: " + e.address);
    System.out.println("SSN: " + e.SSN);
    System.out.println("Number: " + e.number);
  }
}
```

Output:



Example # 06 (Another Serialization Example)

PersonInfo.java

```
import javax.swing.*;
import java.io.*;

class PersonInfo implements Serializable{
String name;
String address;
String phoneNum;

//parameterized constructor
public PersonInfo(String n, String a, String p) {
name = n;
address = a;
phoneNum = p;
}

//method for displaying person record on GUI
public void printPersonInfo() {
JOptionPane.showMessageDialog(null,"name: " + name + "address:" +address + "phone no:"
}

} // end class
```

WriteEx.java (Write an object of person object in file)

```
import java.io.*;

public class WriteEx{

public static void main(String args[ ]){
PersonInfo pWrite =new PersonInfo("ali", "Lahore", "123456");

try {

// attaching FileOutput stream with "ali.txt"
FileOutputStream fos =new FileOutputStream("ali.txt");

// attaching ObjectOutput stream over FileOutput// stream
ObjectOutputStream out =new ObjectOutputStream(fos);

//serialization writing object to 'ali.dat'
out.writeObject(pWrite);

// closing streams
out.close();
fos.close();

} catch (Exception ex){
System.out.println(ex);
}

} //end main method

} // end class
```

ReadEx.java (Read an object of person object in file)

```
import java.io.*;

public class ReadEx{

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

//attaching FileInput stream with "ali.txt"
FileInputStream fis = new FileInputStream("ali.txt");

// attaching FileInput stream over ObjectInput stream
ObjectInputStream in = new ObjectInputStream(fis);


//de-serialization reading object from 'ali.txt'
PersonInfo pRead = (PersonInfo) in.readObject();

// calling printPersonInfo method to confirm that object contains same set of values
//before serializatoion
pRead.printPersonInfo();

// closing streams
in.close();
fis.close();
} catch (Exception ex){
System.out.println(ex);}

}// end of main method
} // end class
```

Output: