



FEU INSTITUTE OF TECHNOLOGY

COLLEGE OF COMPUTER STUDENTS
INFORMATION TECHNOLOGY DEPARTMENT

CCS0023L
(Object Oriented Programming)

EXERCISE

10

Creating, Reading and Writing a File Using Input and Output Streams

Student Name / Group Name:	Danica B. Modino
Section:	AGD TXR02
Professor:	Mr. Roman De Angel

I. PROGRAM OUTCOME/S (PO) ADRESSED BY THE LABORATORY EXERCISE

a. Apply knowledge of computing appropriate to the discipline.

II. COURSE LEARNING OUTCOME/S (CLO) ADDRESSED BY LABORATORY EXERCISE

2 Apply knowledge of computing fundamentals and develop computer programs as a result of evaluating possible alternative program constructs that will address the need for automating real life tasks, taking into account the efficiency and effects of their choice of constraints.

III. INTENDED LEARNING OUTCOME/S (ILO) ADDRESSES BY THE LABORATORY EXERCISE

- *Implement Input and Output stream in creating, reading and writing streams*

IV. BACKGROUND INFORMATION

There are generally two types of streams, the character and byte streams. Let us just review the basic difference between the two. Byte streams are file or device abstractions for binary data while character streams are for Unicode characters.

The `InputStream` class is the abstract root class of all input byte streams whereas the `OutputStream` class is the abstract root class of all output byte streams. For character streams, the corresponding superclass of all classes are the `Reader` and the `Writer` class, respectively. Both classes are abstract classes for reading and writing to character streams.

Streams are also categorized on whether they are used for reading or writing to streams. Although it is already quite obvious, allow me to define these types of streams. You are allowed to read from input streams but not write to them. On the other hand, you are allowed to write to output streams but not read from them.

The `InputStream` class and the `Reader` class are the superclasses of all input streams.

The `OutputStream` class and the `Writer` class are the root classes of all output streams.

Input streams are also known as source streams since we get information from these streams. Meanwhile, output streams are also called sink streams.

A node stream is a stream with the basic functionality of reading or writing from a specific location such as a disk or from the network. Types of node streams include files, memory and pipes. Filter streams, on the other hand, are layered onto node streams between threads or processes to provide additional functionalities not found in the node stream by themselves. Adding layers to a node stream is called stream chaining.

V. EXPERIMENTAL PROCEDURE:

1. Create a program that writes an Integer object to a stream and reads this object from the stream.
2. Simple Encryption. Read from a file specified by the user and encrypt the content of the file using a simple shifting technique. Also, ask the user to input the shift size. Output the encrypted message to another file whose filename is also provided by the user.

For example,

Shift size: 1

Message read from the file: Hello

Encrypted message: Ifmmp

VI. QUESTION AND ANSWER:

1. What is the difference between node and filter streams?







Nodes specifically used in different data structures. It holds data and it is usually is linked to either another node or to a NULL. Filter streams are usually used in collections. It is used to select elements passed as arguments.

First Program Code

```
import java.io.FileOutputStream;
import java.io.FileInputStream;
import java.util.Scanner;

public class one {
    public static void main(String[] args) {
        try {
            FileOutputStream os = new FileOutputStream("number.txt"); //creates new
text file
            FileInputStream is = new FileInputStream("number.txt"); //opens the
said file
            Scanner input = new Scanner(System.in);
            System.out.print("Enter your number here: ");
            int num = input.nextInt();
            os.write(num); //writes the integer inside the text file
            System.out.println("Your number has been saved!");
            os.close();
            System.out.println("*****");
            System.out.println("Gathering Data...");
            System.out.print("'" + (int) is.read()); //prints out the text from the
file
            is.close();
            input.close();
        }
        catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

First Program Output

Name	Status	Date modified	Type	Size
 message		24/11/2021 8:23 pm	Text Document	1 KB
 one		25/11/2021 11:17 am	Java Source File	1 KB
 two		25/11/2021 11:19 am	Java Source File	2 KB

```
PS C:\Users\ailar\OneDrive\Mga Dokumento\2nd year 1st sem\OOP\CODES\TSA4> & 'C:\Program Files\Java\jdk-16.0.2\bin\java.exe' '-XX:+ShowCodeDetails InExceptionMessages' '-cp' 'C:\Users\ailar\AppData\Roaming\Code\User\workspaceStorage\edddfa772bdaed7f502c7c53317fa551\redhat_java\jdt_ws\TSA4_66d c1b89\bin' 'one'
```

```
Enter your number here: 65
```









```
Your number has been saved!
```


```
*****
```

```
Gathering Data...
```

```
65
```

```
PS C:\Users\ailar\OneDrive\Mga Dokumento\2nd year 1st sem\OOP\CODES\TSA4> █
```

 message		24/11/2021 8:23 pm	Text Document	1 KB
 number		25/11/2021 11:36 am	Text Document	1 KB
 one		25/11/2021 11:17 am	Java Source File	1 KB
 two		25/11/2021 11:19 am	Java Source File	2 KB

 number - Notepad

File Edit Format View Help

A

Second Program Code

```
import java.io.*;
import java.util.Scanner;
public class two {
    public static void main(String []args)throws Exception{
        try
        {
            Scanner input=new Scanner(System.in);
            //variables
            String text, out;
            int shift;
            char car;

            //asking user which file and size of shift
            System.out.print("Input the title of the file you want to encrypt: ");
            text=input.next();
            System.out.print("Shift size: ");
            shift=input.nextInt();
            System.out.print("Message read from the file: ");

            //getting the text inside the file
            FileInputStream is = new FileInputStream(text);
            BufferedReader os = new BufferedReader(new InputStreamReader(is));
            out = os.readLine();
            System.out.println(out);

            System.out.print("Encrypted message: ");
            int lnt = out.length(); //size of the text inside the file
            int asc[] = new int[lnt]; //stores the ascii of the text
            int asc1[] = new int[lnt]; //stores the ascii of the shifted text
            char message; //stores the encrypted text

            for (int i = 0; i<lnt; i++)
            {
                car = out.charAt(i); //stores each element on the variable car
                asc[i] = (int) car; //translate the said element to ascii
                asc1[i] = asc[i]+shift; //shifts ascii
                message =(char) asc1[i]; //translates the ascii to char
                System.out.print(message); //print outs the encrypted texts
            }
            os.close();
            is.close();
            input.close();
        }

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

Second Program Output

```
PS C:\Users\ailar\OneDrive\Mga Dokumento\2nd year 1st sem\OOP\CODS\TSA4> c:; cd 'c:\Users\ai
S\TSA4'; & 'C:\Program Files\Java\jdk-16.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMes
er\workspaceStorage\edddfa772bdaed7f502c7c53317fa551\redhat.java\jdt_ws\TSA4_66dc1b89\bin' 'tw
Input the title of the file you want to encrypt: message.txt
Shift size: 2
Message read from the file: Hello
Encrypted message: Jgnnq
PS C:\Users\ailar\OneDrive\Mga Dokumento\2nd year 1st sem\OOP\CODS\TSA4> █
```

Note: The following rubrics/metrics will be used to grade students' output in the lab Exercise 10.

Program (100 pts.)	(Excellent)	(Good)	(Fair)	(Poor)
Program execution (20pts)	Program executes correctly with no syntax or runtime errors (18-20pts)	Program executes with less than 3 errors (15-17pts)	Program executes with more than 3 errors (12-14pts)	Program does not execute (10-11pts)
Correct output (20pts)	Program displays correct output with no errors (18-20pts)	Output has minor errors (15-17pts)	Output has multiple errors (12-14pts)	Output is incorrect (10-11pts)
Design of output (10pts)	Program displays more than expected (10pts)	Program displays minimally expected output (8-9pts)	Program does not display the required output (6-7pts)	Output is poorly designed (5pts)
Design of logic (20pts)	Program is logically well designed (18-20pts)	Program has slight logic errors that do no significantly affect the results (15-17pts)	Program has significant logic errors (3-5pts)	Program is incorrect (10-11pts)
Standards (20pts)	Program code is stylistically well designed (18-20pts)	Few inappropriate design choices (i.e. poor variable names, improper indentation) (15-17pts)	Several inappropriate design choices (i.e. poor variable names, improper indentation) (12-14pts)	Program is poorly written (10-11pts)
Delivery (10pts)	The program was delivered on time. (10pts)	The program was delivered a day after the deadline. (8-9pts)	The program was delivered two days after the deadline. (6-7pts)	The program was delivered more than two days after the deadline. (5pts)

Topic	Input and Output Stream
Lab Activity No	10a
Lab Activity	Writing Int Object in a File
CLO	2
Program execution (20)	
Correct output (20)	
Design of output (10)	
Design of logic (20)	
Standards (20)	
Delivery (10)	
TOTAL	

Topic	Input and Output Stream
Lab Activity No	10b
Lab Activity	Simple Encryption
CLO	2
Program execution (20)	
Correct output (20)	
Design of output (10)	

Design of logic (20)	
Standards (20)	
Delivery (10)	
TOTAL	