

# Exercises: Streams, Files and Directories

This document defines the exercises for ["Java Advanced" course @ Software University](#). Please submit your solutions (source code) of all below described problems in [Judge](#).

For these exercises you are given a zipped folder with resources, that you will need to use. For each exercise submit the output of the program, not the code.

## 1. Sum Lines

Write a program that reads a text file (**input.txt** from the Resources - Exercises) and prints on the console the **sum** of the ASCII symbols of each of its lines. Use **BufferedReader** in combination with **FileReader**.

### Examples

Input	Output
On January 1 , 1533 ,	1452
Michael Angelo,	1397
then fifty-seven years old,	2606
writes	670
from Florence to	1573
Tommaso de' Cavalieri,	2028
a youth of noble Roman family,	2762

### Hints

- Use **try-with-resources** to handle file

```
try () {  
  
} catch (IOException e) {  
    e.printStackTrace();  
}
```

- Create a **BufferedReader** to read each line of the file

```
try (BufferedReader reader = Files.newBufferedReader(Paths.get(inputPath))) {  
  
} catch (IOException e) {  
    e.printStackTrace();  
}
```

- Get the ASCII code of each character in the line and **add** it to the **sum** for the current line and print the sum on the console

```
String line = reader.readLine();

while (line != null) {
    long sum = 0;
    for (char c : line.toCharArray()) {
        sum += c;
    }

    System.out.println(sum);
    line = reader.readLine();
}
```

## 2. Sum Bytes

Write a program that reads a text file (**input.txt** from the Resources - Exercises) and prints on the console the **sum** of the ASCII symbols of all characters inside of the file. Use **BufferedReader** in combination with **FileReader**.

Input	Output
On January 1 , 1533 , Michael Angelo, then fifty-seven years old, writes from Florence to Tommaso de' Cavalieri, a youth of noble Roman family,	12488

### Hints

- You can modify your solution to the previous problem
- Use a type that will not overflow like **long** or **BigInteger**

## 3. ALL CAPITALS!

Write a program that reads a text file (**input.txt** from the Resources - Exercises) and changes the casing of **all** letters to **upper**. Write the output to another file (**output.txt**).

### Examples

Input	Output
On January 1 , 1533 , Michael Angelo, then fifty-seven years old, writes from Florence to Tommaso de' Cavalieri, a youth of noble Roman family,	ON JANUARY 1 , 1533 , MICHAEL ANGELO, THEN FIFTY-SEVEN YEARS OLD, WRITES FROM FLORENCE TO TOMMASO DE' CAVALIERI, A YOUTH OF NOBLE ROMAN FAMILY,

### Hints

- Use **BufferedReader** and **PrintWriter**.

## 4. Count Character Types

Write a program that reads a list of words from the file (**input.txt** from the Resources - Exercises) and finds the count of **vowels**, **consonants** and **punctuation** marks. Assume that:

- **a, e, i, o, u** are vowels, only lower case
- **All others** are consonants
- Punctuation marks are (**!,.,?**)
- **Do not** count whitespace.

Write the results to another file – **output.txt**.

### Examples

Input	Output
On January 1 , 1533 , Michael Angelo, then fifty-seven years old, writes from Florence to Tommaso de' Cavalieri, a youth of noble Roman family,	Vowels: 41 Consonants: 72 Punctuation: 6

### Hints

- Use **BufferedReader** and **PrintWriter**.

## 5. Line Numbers

Write a program that reads a text file (**inputLineNumbers.txt** from the Resources - Exercises) and **inserts** line numbers in front of each of its lines. The result should be written to **another** text file – **output.txt**.

### Examples

Input	Output
Two households, both alike in dignity, In fair Verona, where we lay our scene, From ancient grudge break to new, Where civil blood makes civil hands. From forth the fatal loins of these two A pair of star-cross'd lovers take their life; Whose misadventured piteous overthrows Do with their death bury their parents' strife.	1. Two households, both alike in dignity, 2. In fair Verona, where we lay our scene, 3. From ancient grudge break to new, 4. Where civil blood makes civil hands. 5. From forth the fatal loins of these two 6. A pair of star-cross'd lovers take their life; 7. Whose misadventured piteous overthrows 8. Do with their death bury their parents' strife.

## 6. Word Count

Write a program that reads a list of words from the file **words.txt** (from the Resources - Exercises) and finds how many times each of the words is **contained** in another file **text.txt** (from the Resources – Exercises). Matching should be case-**insensitive**.

Write the results in file **results.txt**. Sort the words by frequency in **descending order**.

Input	Output
of which The	of - 6 which - 2 The - 1

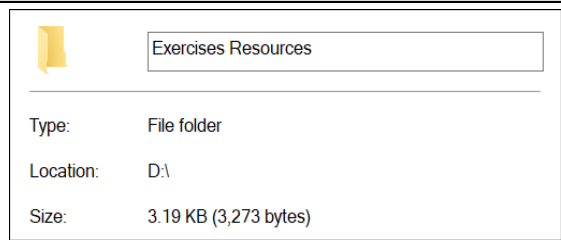
## 7. Merge Two Files

Write a program that reads the contents of **two** text files (**inputOne.txt**, **inputTwo.txt** from Resources Exercises) and **merges** them together into a third one.

File 1	File 2	Output
1	4	1
2	5	2
3	6	3
		4
		5
		6

## 8. Get Folder Size

Write a program that **traverses** a folder and **calculates** its size in bytes. Use Folder **Exercises Resources** in Resources.

Input	Output
	Folder size: 2878

## 9. Copy a Picture

Write a program that makes a copy of a .jpg file using **FileInputStream**, **FileOutputStream**, and **byte[] buffer**. Set the name of the new file as **picture-copy.jpg**.

## 10. Serialize Array List

Write a program that saves and loads an **ArrayList** of doubles to a file using **ObjectInputStream** and **ObjectOutputStream**. Set the name of the file as **list.ser**.

## 11. \*Serialize Custom Object

Write a program that saves and loads the information about a custom object using **ObjectInputStream** and **ObjectOutputStream**.

Create a **simple class** called "Course" that has a **String field** containing its **name** and an **integer field** containing the **number of students** attending the course. Set the name of the save file as **course.ser**.

## 12. \*Create Zip Archive

Write a program that reads three **.txt** files and creates a zip archive named **files.zip**. Use **FileOutputStream**, **ZipOutputStream**, and **FileInputStream**.