

Fundamental Input practice

It is essential to be able to write a computer program to read basic numeric and text input. This exercise will give you some practice with interactive input, file input, tokenizing strings, and exceptions that might occur when we do basic input.

In most computer programming, the fundamental unit of a text file is a single line of text. When we read a file, we are usually interested in reading all of its individual lines, one at a time. This needs to be done in some kind of loop. And then, for each line that we encounter, we may have several items to look for. If the line has a lot of data, or words, it may be necessary to have an inner loop to process each token on the line.

This program that you will write in this exercise will give you practice with some bread and butter operations that are very common in simple Python or Java programs. The specific application is as follows: we wish to read a text file, and count how many words it has. (By “word”, I simply mean any token that is delimited from other tokens by one or more spaces.) Also, it will find the sum of all the integer tokens that it finds in the file.

Let me start you off with some skeleton code in both Python and Java, and you can finish the program. Even if reading text files is a common job for you, you probably don’t want to memorize all the necessary code to process text files. So, use this handout or the program that you write as a reference.

Python skeleton:

```
# input.py - Let's practice input in Python
import re
filename = input("What is the name of the input file? ")
inFile = open(filename, "r")

for line in inFile:
    tok = re.compile(" ")
    tokenlist = tok.split(line)
    tokenlist = [s for s in tokenlist if len(s) > 0]

    for token in tokenlist:
        # You ought to make this a try-except because
        # many tokens are not integers!
        value = int(token)
```

Java skeleton:

```
import java.util.Scanner;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.util.StringTokenizer;

/** Input.java - Practice simple input in Java
 */
public class Input
{
    public static void main(String [] args) throws FileNotFoundException
    {
        Scanner kbd = new Scanner(System.in);
        System.out.printf("What is the name of the input file? ");
        String filename = kbd.nextLine().trim();
        Scanner in = new Scanner(new FileInputStream(filename));

        while(in.hasNextLine())
        {
            String line = in.nextLine();
            StringTokenizer tok = new StringTokenizer(line, " ");

            while (tok.hasMoreTokens())
            {
                // You need a try/catch because most tokens are not numbers!
                int value = Integer.parseInt(tok.nextToken());
            }
        }
    }
}
```

1. Your program should conclude by closing the input file, and then printing the number of words and the integer sum. Run your program on an example input file!
2. Modify your program so that it also finds the longest word (i.e. token) in the file.
3. Modify your program to guard against the possibility that the file name that the user enters is invalid. If so, give the user another chance. To do this you need to enclose the code that takes the file name inside a loop, and you need to insert a try/catch block (try/except in Python).