UNIVERSITY OF KWAZULU-NATAL

COMP102: Compute Programming

Practical 8: File Processing

Thursday, 13 October 2022

Question 1: Text Analysis

One way to determine the difficulty of reading a text is to determine the average length of words in the text.

Write a program that analyses the text file **textProcessing.txt**. The text file contains a number of lines of text.

Your program should output:

- the number of lines of text
- the number of words in the file
- the average word length of all the words in the file. Get the name of the file to process from the user

Use the following template to report your analysis:

```
There are x lines of text in the file.
There are x words in the file.
The average number length of words is x.
```

Question 2: Separate Data

The text file **separateData.txt** contains marks for three students over the course of the year. The first three lines contain the names of the students (surname only). The rest of the file contains on each line, a name of one of the students followed by a space and a mark. Nothing else follows the mark on the line. The order of the entries follows no pattern.

Write a program that reads in this data file and:

- writes all marks belonging to a particular student to its own file i.e. you output three files with each file containing the marks of a one student.
- computes and outputs to the screen, the average mark for each student.

The following three text files should be created:

1. Gouws.txt
2. Moodley.txt

3. Kingu.txt

Question 3: Email Address Extractor

Write a program that scans a text file for possible e-mail addresses and writes all email address to an output file.

Addresses look like this:

- `someone@somewhere.net`

Notice that the address contains no spaces and has an @ sign followed by at least one period (.). Programs such as this scan through web pages looking for e-mail addresses that become the targets of spam. Because of this, many web pages contain disguised e-mail addresses that can't easily be automatically extracted.

Your program should find 12 email addresses in the text file **emailText.txt**.

Question 4: Check Braces

A simple syntax check of java files is to count the number of open and closed brackets. If the numbers are not the same, then a bracket is missing. Write a program that reads a **.java** file specified by the user and counts the number of open brackets (`{`) and the number of closed brackets (`}`). It should then print out the number of brackets missing. Use the file **Athlete.java** to test.

The braces in the **Althlete.java** file match perfectly. So, you should have equal counts for opening and closing braces. You can then add and remove them from the file to check if your program works as expected.

Question 5: Header Extraction

Write a program that reads a **.java** file specified by the user and prints all the method headers only to a file called **headers.txt**. Use the file **Athlete.java** to test.

A method header is for example:

```
public void getNumber(int i).
```

Your code should write the following headers to the file **headers.txt**.

```
public class Athlete {
public Athlete(String name,int age) {
public String getName() {
public int getAge() {
public double getBestThrow () {
public String toString(){
private double getMinIndex(){
public void update(double newThrow){
```

-