CSC272 Advanced Programming in Java

**Midterm**

|  |  |
| --- | --- |
| First Name | Asa Jean |
| Last Name | Soriano |
| ID# | 041702547 |

Table of Contents

[How to submit your Midterm Exam 1](#_Toc53122538)

[P1 – Word Frequency Count (25 points) 2](#_Toc53122539)

[P2 – File Handler (25 points) 3](#_Toc53122540)

[P3 – The Array Multiplier (25 Points) 4](#_Toc53122541)

[P4 – HashMap Histogram (25 Points) 5](#_Toc53122542)

# How to submit your Midterm Exam

After filling all the parts in this file, please follow the following steps.

1. Add your name and ID to the first page.
2. Save the file in the original format (Docx or Doc)

(please do not convert to other file formats e.g. PDF, ZIP, RAR, …).

1. Rename the file as

*YOUR* ***First*** *Name - YOUR* ***Last*** *Name -* CSC272 *–* ***Midterm*** *- ID.docx*

**Example:**

John – Smith - CSC272 *–* ***Midterm*** *-* 234566435.docx

1. Upload the file and submit it (only using Blackboard)

# P1 – Word Frequency Count (25 points)

Write a program that reads a line of text and prints a table indicating the number of occurrences of each different word in the text. The application should include the words in the table in the same order in which they appear in the text. The application should also ignore the following common punctuation marks: **, . : ; ? ! “**   
For example, the lines

***To be, or not to be: that is the question: Whether ‘tis nobler in the mind to suffer***

contain the word “to” three times, the word “be” two times, the word “or” once etc.

Results:

Word Count

to 3

be 1

or 1

not 1

be 1

that 1

is 1

the 2

question 1

whether 1

'tis 1

nobler 1

in 1

mind 1

suffer 1

|  |
| --- |
| Your code for this problem |
| import java.util.HashMap;  import java.util.Scanner;  public class P1 {  // P1 - Word Frequency Count  // program that reads line of text from user input and prints a table of the number of occurrences for each word in the text and ignores punctuation  //The table where the key is the word (String) and the value is the count (Integer)  static HashMap<String, Integer> wordCount = new HashMap<String, Integer>();  //method to take a String and put each unique word in table "wordCount" and increase the value of each word by 1 if it's already in the table  public static void putWords(String sentence) {  //split on punctuation and white spaces  String[] words = sentence.split("[\\p{Punct}\\s]+");  for (String word : words) {  word = word.toLowerCase();  if (wordCount.containsKey(word)) {  wordCount.put(word, wordCount.get(word) + 1);  } else {  wordCount.put(word, 1);  }  }  }  public static void main (String[] args) {  //take string from input  System.out.println("Enter your sentence.");  Scanner in = new Scanner(System.in);  String sentence = in.nextLine();  putWords(sentence);  //print the wordCount table  System.out.println("Word\t\tCount");  for (String word : wordCount.keySet()) {  System.out.println(word + "\t\t" + wordCount.get(word));  }  in.close();  }  } |

Run the code for the string given in the problem statement and another string of your choice and insert the results in the following box.

|  |
| --- |
| The result of the query |
|  |

# P2 – File Handler (25 points)

Write a program, using a Formatter object to write to a file, that does the following:  
Part (a) – Generate 10,000 random integer numbers between 0 and 9 including both 0 and 9, and write all the random integer, one number per line, to an output text file “numbers.txt”.

Part (b) -- Upon completion of part (a), read all the integers from the file “numbers.txt” generated from part (a), calculate the frequency of occurrence for each of the different numbers, and output the result in the following format to another text file “output.txt”

Output.txt format:

Numbers Frequency  
number 1 frequency of number 1  
number 2 frequency of number 2

…

Important: In both part (a) and part(b), please properly handle some or all of the following exceptions when they apply and properly close all files before program terminates.

SecurityException  
FileNotFoundException   
IllegalFormatException   
FormatterClosedException

NoSuchElementException

IllegalStateException

|  |
| --- |
| Your code for this problem |
| import java.util.Formatter;  import java.util.HashMap;  import java.util.Scanner;  import java.security.SecureRandom;  import java.io.File;  public class P2 {  public static void main (String[] args) {  final int SIZE = 10000;  //generate 10000 random integers between 0 and 9 including both 0 and 9  SecureRandom randomNumbers = new SecureRandom();  //write each random number to file numbers.txt  File file = new File("numbers.txt");  try {  Formatter output = new Formatter(file);  for (int i = 0; i < SIZE; i++) {  output.format("%d\n", randomNumbers.nextInt(10));  }  output.close();  } catch (Exception e) { //catch any exceptions  System.out.println("Error: " + e.getMessage());  }  //read integers from numbers.txt and count frequency of each unique number  HashMap<Integer, Integer> numberCount = new HashMap<Integer, Integer>();  try {  Scanner input = new Scanner(file);  while (input.hasNext()) {  int number = input.nextInt();  if (numberCount.containsKey(number)) {  numberCount.put(number, numberCount.get(number) + 1);  } else {  numberCount.put(number, 1);  }  }  input.close();  } catch (Exception e) {  System.out.println("Error: " + e.getMessage());  }  File out = new File("output.txt");  //put numberCount in output.txt  try {  Formatter output = new Formatter(out);  output.format("Number\t\tCount\n");  for (int i = 0; i < 10; i++) {  output.format("%d\t\t%d\n", i, numberCount.get(i));  }  output.close();  } catch (Exception e) {  System.out.println("Error: " + e.getMessage());  }  //print the numberCount table  System.out.println("Number\t\tCount");  for (int n : numberCount.keySet()) {  System.out.printf("%d\t\t%d\n", n, numberCount.get(n));  }  }    } |

Run the code and insert the contents of your output.txt in the following box. Also submit your numbers.txt and output.txt along with this midterm exam document.

|  |
| --- |
| The result of the query |
|  |

# P3 – The Array Multiplier (25 Points)

Given the definition of a 2D array such as the following:

String[][] data = { {"A","B"}, {"1","2"}, {"XX","YY"} };

Write a recursive program that outputs all combinations of each subarray in order. In the above example, the desired output (although it doesn’t have to be in this order) is:

A 1 XX  
 A 1 YY  
 A 2 XX  
 A 2 YY  
 B 1 XX  
 B 1 YY  
 B 2 XX  
 B 2 YY

Your program should work with arbitrarily sized arrays in either dimension. For example, the following data:

String[][] data = { {"A"}, {"1"}, {"2"}, {"XX","YY"} };

should output:

A 1 2 XX  
 A 1 2 YY

|  |
| --- |
| Your code for this problem |
| public class P3 {  public static void main (String[] args) {  //The Array Multiplier  //a recursive program that outputs all combinations of each subarray in order given a 2D array  //String[][] data = { {"A", "B"}, {"1", "2"}, {"XX", "YY"} };  //String[][] data = { {"A"}, {"B"}, {"1"}, {"2"}, {"XX", "YY"} };  String[][] data = {{"Try","Do"}, {"at least"}, {"1","3"}, {"Tasks", "Activity"}, {"a day","a week"} };  StringBuilder sb = new StringBuilder();  printCombinations(data, 0, 0, sb);  }  //recursive method to print all combinations of each subarray in order given a 2D array  public static void printCombinations(String[][] data, int row, int col, StringBuilder sb) {  if (row == data.length) {  System.out.println(sb.toString());  return;  }  for (int i = 0; i < data[row].length; i++) {  sb.append(data[row][i] + " ");  printCombinations(data, row + 1, i, sb);  sb.delete(sb.length() - data[row][i].length() - 1, sb.length());  }  }  } |

Run the code for the following 2D array:

String[][] data = {{“Try”,”Do”}, {“at least”}, {“1”,”3”}, {“Tasks”, “Activity”}, {“a day”,”a week”} };

|  |
| --- |
| The result of the query |
|  |

# P4 – HashMap Histogram (25 Points)

Write a program that uses a HashMap to compute a histogram of positive numbers entered by the user. The HashMap’s key should be the number that is entered, and the value should be a counter of the number of times the key has been entered so far. Use −1 as a sentinel value to signal the end of user input. For example, if the user inputs:

5  
12  
3  
5  
5  
3  
21  
−1

then the program should output the following (not necessarily in this order):

The number 3 occurs 2 times.  
The number 5 occurs 3 times.  
The number 12 occurs 1 times.  
The number 21 occurs 1 times.

In your program, also design a function called **printHistogram** that takes the HashMap histogram and display it in the above format.

|  |
| --- |
| Your code for this problem |
| import java.util.HashMap;  import java.util.Scanner;  public class P4 {  public static void main (String[] args) {  //Hashmap Histogram  //create a HashMap to compute histogram of positive numbers from user input, where the key is the number and the value is the frequency    HashMap<Integer, Integer> histogram = new HashMap<Integer, Integer>();  Scanner in = new Scanner(System.in);  while (in.hasNext()) {  int number = in.nextInt();  if (number == -1) {  break;  }  if (histogram.containsKey(number)) {  histogram.put(number, histogram.get(number) + 1);  } else {  histogram.put(number, 1);  }  }  in.close();  printHistogram(histogram);  }  public static void printHistogram(HashMap<Integer, Integer> histogram) {  //print the histogram  for (int number : histogram.keySet()) {  System.out.printf("The number %d occurs %d times.\n", number, histogram.get(number));  }  }    } |

Run the code and insert the result in the following box.

|  |
| --- |
| The result of the query |
|  |