**CSC272 Advanced Programming in Java**

**Assignment 1**

|  |  |
| --- | --- |
| **First Name** | Asa Jean |
| **Last Name** | Soriano |
| **ID#** | 041702547 |
| **Assignment Week#** | Week 1 |
| **Email Address** | asajeansoriano@gmail.com |

Contents

[How to submit your Assignment 1](#_Toc52034694)

[Problem 1 – *Metric Converter* 2](#_Toc52034695)

[Problem 2 – *Telephone Number Word Generator* 2](#_Toc52034696)

# How to submit your Assignment

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- YOUR student ID-* CSC272*.docx*

**Example:**

John – Smith - 234566435 – CSC272.docx

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

Please do not hesitate to contact me, should you have any questions.

# Problem 1 – *Metric Converter*

Write an application that will assist the user with metric conversions. Your application should allow the user to specify the names of the units as strings (i.e., centimeters, liters, grams, and so on, for the metric system and inches, quarts, pounds, and so on, for the English system) and should respond to simple questions, such as

"How many inches are in 2 meters?"

"How many liters are in 10 quarts?"

Your application should recognize invalid conversions. For example, the question

"How many feet are in 5 kilograms?"

is not meaningful because "feet" is a unit of length, whereas "kilograms" is a unit of mass.

|  |
| --- |
| Your code for this problem |
|  |

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

|  |
| --- |
| Sample Run Result |
| Fail states |

# Problem 2 – *Telephone Number Word Generator*

Standard telephone keypads contain the digits zero through nine. The numbers two through nine each have three letters associated with them in the following figure. Many people find it difficult to memorize phone numbers, so they use the correspondence between digits and letters to develop seven-letter words that correspond to their phone numbers. For example, a person whose telephone number is 686-2377 might use the correspondence indicated in the following figure to develop the seven-letter word “NUMBERS.” Every seven-letter word corresponds to exactly one seven-digit telephone number. A restaurant wishing to increase its takeout business could surely do so with the number 825-3688 (i.e., “TAKEOUT”).

Every seven-letter phone number corresponds to many different seven-letter words, but most of these words represent unrecognizable juxtapositions of letters. It’s possible, however, that the owner of a barbershop would be pleased to know that the shop’s telephone number, 424-7288, corresponds to “HAIRCUT.” A veterinarian with the phone number 738-2273 would be pleased to know that the number corresponds to the letters “PETCARE.” An automotive dealership would be pleased to know that the dealership number, 639-2277, corresponds to “NEWCARS.”

| **Digit** | **Letters** | **Digit** | **Letters** | **Digit** | **Letters** |
| --- | --- | --- | --- | --- | --- |
| 2 | A B C | 5 | J K L | 8 | T U V |
| 3 | D E F | 6 | M N O | 9 | W X Y |
| 4 | G H I | 7 | P R S |  |  |

Telephone keypad digits and letters.

Write a program that, given a seven-digit number, uses a Formatter object to write to a file every possible seven-letter word combination corresponding to that number. There are 2,187 (37) such combinations. Avoid phone numbers with the digits 0 and 1.

|  |
| --- |
| Your code for this problem |
| import java.io.File;  import java.util.Formatter;  import java.util.HashMap;  import java.util.Scanner;  public class P2 {  //Every seven-letter word corresponds to exactly one seven-digit telephone number  //e.g. 825-3688 is "TAKEOUT"  //424-7288 is "HAIRCUT"  //738-2273 is "PETCARE"  //639-2277 is "NEWCARS"  //This program, given a seven-digit number, uses a Formatter object to write to a file every possible seven-letter word combination corresponding to that number  public static void main(String[] args) {  //keypad arraylists where the digit is the key and the value are the letters  HashMap<Integer, String> keypad = new HashMap<Integer, String>();  keypad.put(2, "ABC");  keypad.put(3, "DEF");  keypad.put(4, "GHI");  keypad.put(5, "JKL");  keypad.put(6, "MNO");  keypad.put(7, "PQR");  keypad.put(8, "TUV");  keypad.put(9, "WXY");  String permutation = "";  String combinations[] = new String[2187];  int ctr = 0;  //find every permutation of 0, 1, 2 in 7 seven numbers  //the 0, 1, 2 represent the corresponding keypad value  //e.g. "ABC" is "012"  for (int i = 0; i < 3; i++) {  for (int j = 0; j < 3; j++) {  for (int k = 0; k < 3; k++) {  for (int l = 0; l < 3; l++) {  for (int m = 0; m < 3; m++) {  for (int n = 0; n < 3; n++) {  for (int o = 0; o < 3; o++) {  permutation = String.valueOf(i) + String.valueOf(j) + String.valueOf(k) + String.valueOf(l) + String.valueOf(m) + String.valueOf(n) + String.valueOf(o);  combinations[ctr++] = permutation;  }  }  }  }  }  }  }    //write to String output  String output = "";  //read in the seven-digit number  Scanner in = new Scanner(System.in);  System.out.println("Enter a seven-digit number: ");  String number = in.nextLine();  //iterate through combinations list and match the number to the keypad  for (int i = 0; i < 2187; i++) {  String temp = combinations[i];  for (int j = 0; j < 7; j++) {  int temp\_index = Character.getNumericValue(temp.charAt(j));  int number\_index = Character.getNumericValue(number.charAt(j));  //split keypad value and match to the corresponding digit  String keypad\_value[] = keypad.get(number\_index).split("");  output += keypad\_value[temp\_index];  }  output += "\n";  }  //System.out.println(output);  //write output to file  File file = new File("output.txt");  try {  System.out.println("Writing to file...");  Formatter output\_file = new Formatter(file);  output\_file.format(output);  output\_file.close();  } catch (Exception e) {  System.out.println("Error writing to file");  }  }  } |
|  |

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

|  |
| --- |
| Sample Run Result |
|  |

**The end**