**Group Project 1 Analysis and Design**

**Analysis:**

The client/user will be able to provide a string of up to 7 letters into the program via the console when prompted. The program will convert any letters into numbers and print the result back to the console. Letters are converted as follows:

* A, B, or C 🡪 2
* D, E, or F 🡪 3
* G, H, or I 🡪 4
* J, K, or L 🡪 5
* M, N, or O 🡪 6
* P, Q, R, or S 🡪 7
* T, U, or V 🡪 8
* W, X, Y, or Z 🡪 9

No other symbols are allowed other than alphabetic letters and the phrase must not exceed 7 characters.

**Design:**

Main() {

Declare a bool (keepGoing), char and string (letters)

keepGoing = true

printTitle()

while(keepGoing) // allows user to continue converting strings until done

print “Do you wish to convert some letters? “

char = getYorN()

If yes

Print “what letters do you want to convert? “

getString()

print “Your number is: “ + convertString(string letters)

// Conversion function should return an integer and the main should print it.

If no

keepGoing = false // break out of loop

print “thank you for using my program.”

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

Pre-condition: none

Post-condition: prints title and program description to console

Void printTitle() {

Print title and description of program using System.out.println()

}

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

Pre-condition: none

Post-condition: uses input.nextLine()

Returns a string of length 7 and which contains only lower case letters

String getAlphaStringLen7()

declare string

bool badInput = false // to handle non-alphabet input

string = getStringLen7()

for (int i = 0; i < string.length AND not badInput; i++)

if (string.charAt(i) is not a letter)

badInput = true

while (badInput)

print “Please use alphabet letters only. Please try again: ”

string = getStringLen7()

Return string

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

Pre-condition: none

Post-condition: uses input.nextLine()

Returns a trimmed string of length 7

String getStringLen7()

declare and init string and scanner

string = scanner.nextLine

string.trim // prevents whitespace from invalidating input

while (string.length not 7) {

print “Please choose 7 letters. Please try again: ”

scan string

Return string.toLowerCase // reduces work of letter validation

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

Pre-condition: none

Post-condition: uses input.nextLine()

Returns ‘y’ or ‘n’

Char getYorN() {

declare and init string, char, and scanner

string = scanner.nextLine

string.toLowerCase

char = string.charAt(0)

while (not ‘y’ or not ‘n’)

print “please enter y or n: ”

scan again

If (‘y’ or ‘Y’)

return ‘y’

else

return ‘n’

}

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

Pre-condition: letters.length is 7

letters is only lowercase letters

Post-condition: returns an integer 7 digits long

Each digit and corresponding char from letters can be found together on dialpad

int convertString(string letters)

declare a char and an int (num) variable

num = 0

for (i < letters.length)

switch (letters.charAt(i))

case ‘A’:

case ‘B’:

case ‘C’:

num += 2 // number that matches letter on dialpad

num \*= 10 // append 0 to make room for next converted letter

break

// repeat pattern for each number on phone

Return num / 10 // the last zero we add is extra