Object Oriented Programming & Design
CS 244 - 001
Department of Physics and Computer Science
Medgar Evers College
Exam 1: Part B

Instructions:

- The exam requires completing a set of tasks within 50 minutes or until the end of class.
- Write your solution to sections III - IV in the Exam01 directory of your GitHub repository.
- Notes are not allowed.
- Cheating of any kind is prohibited and will not be tolerated.
- Violating and/or failing to follow any of the rules will result in an automatic zero (0) for the exam.

Section III

To avoid misspelling a word during radio communications, call signs, which are word associations to the letters in the alphabet, are used. Given that the call signs used in the military are

A - ALFA	B - BRAVO
C - CHARLIE	D - DELTA
E - ECHO	F - FOXTROT
G - GOLF	H - HOTEL
I - INDIA	J - JULIETT
K - KILO	L - LIMA
M - MIKE	N - NOVEMBER
O - OSCAR	P - PAPA
Q - QUEBEC	R - ROMEO
S - SIERRA	T - TANGO
U - UNIFORM	V - VICTOR
W - WHISKEY	X - X–RAY
Y - YANKEE	Z - ZULU

your objective is to translate the letters of a word to their respective call signs above. More specifically, create a cpp file and define a string function named `Translate()` that takes a string parameter. It must return a string that is a copy of the parameter except that letters are converted to their respective call signs and there is a space between each character [call signs are considered characters]. The function should be case-insensitive and you can only include the libraries *iostream*, *string*, *cctype*. For instance, the callers `Translate("Tr3e")` and `Translate("f@!L")` will return `"TANGO ROMEO 3 ECHO"` and `"FOXTROT @ ! LIMA"` respectively.

Hint: Use an array to hold the call signs, and use `isalpha()` and either `tolower()` or `toupper()`.

Section IV

Create a header file named `"SWITCHBOARD.h"` and define the class *SwitchBoard* within the namespace *EX* with a header guard. The class must contain

- ☐ a public bool array field named *switches* with a size of 10.
- ☐ a public default constructor that assigns false to each element of *switches*.
- ☐ a public copy constructor.
- ☐ a public assignment operator.
- ☐ a public empty destructor.
- ☐ a public int constant method named `TotalOn()` that takes no parameters and returns the total number of elements of *switches* whose value is equal to true.
- ☐ a public int constant method named `TotalOff()` that takes no parameters and returns the total number of elements of *switches* whose value is equal to false.
- ☐ a public void method named `Invert()` that takes no parameters and it inverts the values of each element of *switches* [changes true to false, and vice versa].
- ☐ a public string constant method named `ToString()` that takes no parameters. It returns a string of the values of the elements of *switches* such that each is enclosed in square brackets in order. For instance, the call `SwitchBoard().ToString()` would return `"[0][0][0][0][0][0][0][0][0][0]". [false = 0, true = 1]`
- ☐ an ostream operator that returns an outcome in the same format as `ToString()`.