## COMP 53: Templates Lab

***Instructions:*** In this lab, we are going to review templates in function and class definitions.

- Get into groups of **at most two people** to accomplish this lab.
- At the top of your source code files list the group members as a comment.
- Each member of the group must individually submit the lab in Canvas.
- This lab includes **37 points** in aggregate. The details are given in the following.

# 1 `main.cpp`

In `main.cpp` do the following step by step:

1. Globally define integer array `a[]`, and initialize it with numbers 6,10, -4, 11, and -2.
2. Globally define character array `str[]` of length 7, and initialize it with `"Hello."`.
3. Globally declare `vec1` to be a vector of integers, and `vec2` to be a vector of characters.
4. Define the following functions.
   (a) Define function `void reverseArray(...)` that receives an array of *any arbitrary type*, along with its size (an integer). It reverses the order of array elements ***(2 points)***.
   (b) Define function `void printArray(...)` that receives an array of *any arbitrary type*, along with its size (an integer). It prints out the array elements ***(2 points)***.
   (c) Define function `void initVector(...)` that receives 1) a vector of *any arbitrary type* as a reference, 2) an array of the same type as the elements of the input vector, and 3) the size of the array (an integer). It initializes the input vector with the elements existing in the input array ***(2 points)***.
   (d) Define function `void replaceLastThree(...)` that receives a vector of *any arbitrary type* as a reference and three items of the same type as the elements of the input vector. It replaces the last three elements of the vector with those three input items (in order) ***(3 points)***.
   (e) Define function `void printVector(...)` that receives a vector of *any arbitrary type* and prints out its elements ***(2 points)***.
   (f) Define class `Comparable` using templates which
      (i) has two private data components of the same type, named `item1` and `item2` ***(2 points)***.
      (ii) has public function `bool lt()` that checks if `item1` is less than `item2` ***(2 points)***.
      (iii) has public function `bool gt()` that checks if `item1` is greater than `item2` ***(2 points)***.
      (iv) has public function `bool eq()` that checks if `item1` is equal to `item2` ***(2 points)***.
      (v) has constructor `Comparable(...)` that receives two inputs of the same type as the private data components and assigns the first input to `item1`, and the second input to `item2` ***(2 points)***.

In `main()` function do the following step by step, using the functions defined above:

(I) Call reversing array function from above on `a[]` and the appropriate size ***(1 points)***.

(II) Print out the content of `a[]` using `printArray` function ***(1 points)***.

(III) Call the same reversing array function from above on `str[]` and the appropriate size ***(1 points)***.

(IV) Print out the content of `str[]` using the same `printArray` function ***(1 points)***.

(V) Initialize `vec1` according to `a[]` using the initialization function from above ***(1 points)***.

(VI) Print out the content of `vec1` using `printVector` function ***(1 points)***.

(VII) Initialize `vec2` according to `str[]` using the same initialization function from above ***(1 points)***.

(VIII) Print out the content of `vec2` using the same `printVector` function ***(1 points)***.

(IX) Replace last three elements of `vec1` with 9, 0 , and 2, using the appropriate function from above ***(1 points)***.

(X) Print out the content of `vec1` using `printVector` function ***(1 points)***.

(XI) Replace last three elements of `vec2` with `'f'`, `'y'` , and `'o'`, using the same appropriate function from above ***(1 points)***.

(XII) Print out the content of `vec1` using the same `printVector` function ***(1 points)***.

(XIII) Declare a `Comparable` object `comp1` with initial values `3` and `5` for `item1` and `item2`, respectively ***(1 points)***.

(XIV) Declare a `Comparable` object `comp2` with initial values `'r'` and `'b'` for `item1` and `item2`, respectively ***(1 points)***.

(XV) Run `lt()` for `comp1` and print out the result ***(1 points)***.

(XVI) Run `eq()` for `comp2` and print out the result ***(1 points)***.

The output of the program may look like the following:

```
Reverse a[]:
-2, 11, -4, 10, 6

Reverse str[]:
, ., o, l, l, e, H

Initialize vec1 according to a[]:
-2, 11, -4, 10, 6

Initialize vec2 according to str[]:
, ., o, l, l, e, H

Replace last three elements of vec1 with 9, 0 , and 2:
-2, 11, 9, 0, 2

Replace last three elements of vec2 with 'f', 'y' , and 'o':
, ., o, l, f, y, o

Is 1st component of comp1 less than 2nd component of it? 1
Is 1st component of comp2 equal to 2nd component of it? 0
```