CSCE 221 Cover Page
Programming Assignment #1
Due July 11 by midnight to eCampus

First Name Last Name UIN

User Name E-mail address

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero, read more: Aggie Honor System Office

Type of sources			
People			
Web pages (provide URL)			
Printed material			
Other Sources			

I certify that I have listed all the sources that I used to develop the solutions/codes to the submitted work.

"On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work."

Your Name Date

Programming Assignment 1 (130 points)

In the first phase of the assignment, implement in C++ a class `My_vec` that can hold data of character type (`char`). The description of the functions for data manipulation is provided in the first set of the lecture notes, see the slide 8. In the second phase, write a generic version of the class `My_vec` that can handle any type of data.

Instructions:

1. Download the supplementary file with a sample code from the class webpage.
2. Your files should be arranged as follows
   (a) Declaration of `My_vec` class in `My_vec.h`
   (b) Definition (implementation) of `My_vec` class in `My_vec.cpp`
   (c) Testing code in `Main.cpp`
   (d) Use `Makefile` by calling `make`
3. Compile your program using the Linux machine command line:
   `g++ -std=c++11 *.cpp`
   or
   `make all`
4. Run your program by executing
   `./Main`
5. Be sure to increase or decrease allocated memory when you insert to or remove from a vector.
6. Be sure to check the vector size against its capacity. If its size is greater than its capacity then allocate more memory by doubling the current capacity and copying the content of the vector.

Points Distribution for Assignment (for part 1 and 2)

1. `My_vec` class member functions:
   (a) (4 pt) `elem_at_rank`
   (b) (8 pt) `insert_at_rank`
      i. (4 pt) `replace_at_rank`
      ii. (8 pt) `remove_at_rank`
      iii. (10 pt) constructor and copy constructor
      iv. (10 pt) destructor and the assignment operator
      v. (6 pt) overloading the bracket `[]` operator
   (c) non-member functions:
      i. (6 pt) overloading `<<` operator (output operator)
      ii. (10 pt) `find_max_index` – find an index of the largest object in a vector
      iii. (10 pt) sort `My_vec` using the function `find_max_index`
   (d) (14 pt) Testing program (main function)

   (5 pt) Programming style: naming, indentation, whitespace, comments, declaration, variables and constants, expressions and operators, line length, error handling and reporting. Please refer to the document PPT-style.

2. (10 pt) Generic version of `My_vec`

   (a) The templated "`My_vec`" uses the data type as a parameter. Recall the templated vector material, slides 16-22 and follow the instructions below

      i. Templates should be declared and defined in the `TemplatedMy_vec.h` file. Move the content of `My_vec.cpp` and `My_vec.h` to `TemplatedMy_vec.h`

      ii. Replace `char` type by generic type `T`. Later, in the main function, `T` can be specified as any type: `char`, `string` or a user-defined type.

      iii. To create a templated class with generic type `T`, you must replace a declaration/return type `char` by `T`.

      iv. Use the generic type `T` anywhere throughout the class `TemplatedMy_vec`.

      v. Add the keyword `template <typename T>` before a class declaration.

      vi. If a member function is defined outside the class declaration, change the function signature, that is, replace `My_vec::` by `TemplatedMy_vec<T>::`

   (b) Compile and run the generic version similarly as in the part 1 of the assignment.

   (c) (5 pt) Test all the operations for the generic version using at least three different types of objects.

3. Typed report (preferably using "LyX/LaTeX") – Report Instructions

   (a) (1 point) Cover Page

   (b) (1 pt) Program Description; Purpose of the Assignment

   (c) (4 pt) Data Structures Description

   - Theoretical definition
   - Real implementation
   - Analysis of best and worst scenarios for vector.

   (d) (2 pt) Instructions to Compile and Run your Program; Input and Output Specifications

   (e) (2 pt) Logical Exceptions (and bug description)

   (f) (5 pt) C++ object oriented or generic programming features, C++11 features.

   (g) (5 pt) Testing results

Submission to eCampus no latter than July 11.

Please create a folder for each phase (1 and 2) of the `My_vec` class and compress the two directories into a tar file according to the eCampus instructions.