National University of Computer and Emerging Sciences



Assignment 6

For

Object Oriented Programming

|  |  |
| --- | --- |
| Course Instructor(s) | Mr. Usman Ghous |
| Lab Instructor(s) | Mr. Usman Ghous |
| Semester | Spring 2021 |

**FAST School of Computing**

# Instructions:

1. Make a word document with the naming convention “SECTION\_ LAB#\_ROLLNO” and put all your source code and snapshots of its output in it. Make sure your word file is formatted properly.
2. **Plagiarism is strictly prohibited.**
3. Do not discuss solutions with one another.
4. Feel free to explore the internet but do not copy.
5. Try coming up with a unique solution with whatever you have studied in OOP.

# Useful links

<https://www.geeksforgeeks.org/templates-cpp/>

<https://www.geeksforgeeks.org/class-template-multiple-parameters/>

|  |
| --- |
| **Question#1** |

Write a template function swap to perfrom swap between any two values.

|  |
| --- |
| **Question#2** |

Write a **template** function **sort()** to sort any array. The parameters will be array and its size.

|  |
| --- |
| **Question#3** |

Write a template class **Calculator** which will perform the functionalities of calculator having two private attributes and having functions add, subtract, multiply, divide.

|  |
| --- |
| **Question#4** |

Vectors are same as dynamic arrays with the ability to resize itself automatically when an element is inserted or deleted, with their storage being handled automatically by the container. So, one must attribute is a pointer. You’ll decide the remaining attributes yourself after reading the function you are going to implement. The following functions in your class vector will be:

1. [size()](https://www.geeksforgeeks.org/vectorempty-vectorsize-c-stl/) – Returns the number of elements in the vector.
2. [max\_size()](https://www.geeksforgeeks.org/vector-max_size-function-in-c-stl/) – Returns the maximum number of elements that the vector can hold.
3. [resize(n)](https://www.geeksforgeeks.org/vector-resize-c-stl/) – Resizes the container so that it contains ‘n’ elements.
4. [empty()](https://www.geeksforgeeks.org/vectorempty-vectorsize-c-stl/) – Returns whether the container is empty.
5. [at(g)](https://www.geeksforgeeks.org/vectorat-vectorswap-c-stl/) – Returns a reference to the element at position ‘g’ in the vector
6. [front()](https://www.geeksforgeeks.org/vectorfront-vectorback-c-stl/) – Returns a reference to the first element in the vector
7. [back()](https://www.geeksforgeeks.org/vectorfront-vectorback-c-stl/) – Returns a reference to the last element in the vector
8. [assign()](https://www.geeksforgeeks.org/vector-assign-in-c-stl/)– It assigns new value to the vector elements by replacing old ones
9. [push\_back()](https://www.geeksforgeeks.org/vectorpush_back-vectorpop_back-c-stl/) – It push the elements into a vector from the back
10. [pop\_back()](https://www.geeksforgeeks.org/vectorpush_back-vectorpop_back-c-stl/) – It is used to pop or remove elements from a vector from the back.
11. [insert()](https://www.geeksforgeeks.org/vector-insert-function-in-c-stl/) – It inserts new elements before the element at the specified position
12. [mplace()](https://www.geeksforgeeks.org/vector-emplace-function-in-c-stl/) – It extends the container by inserting new element at position.
13. Overload operator << which displays all of the inserted elements.

There are many function of vectors, but you are only going to implement few of them.

You can start your program by sending an integer in constructor to create an array of passed size. The resizing will happen automatically, implement any other function inside the class if required. You can take help from the given link to understand vectors more deeply <https://www.geeksforgeeks.org/vector-in-cpp-stl/> .**(Provide default, copy constructors and a destructor to destroy the allocated memory.)** Test all functions with 2 or 3 data types.

|  |
| --- |
| **Question#5** |

Write a class **Calculator** which will perform the functionalities of calculator having two private attributes and having functions add, subtract, multiply, divide. Use exception handling for validating input for all possible cases.

|  |
| --- |
| **Question#6** |

Write a program illustrating that all destructors for objects constructed in a block are called before an exception is thrown from that block.

|  |
| --- |
| **Question#7** |

Write a program that throws an exception from a deeply nested function and still has the catch handler following the try block enclosing the initial call in main catch the exception.