ECE 1410 Generic Stacks and Queues Requirements

Stacks and queues are important data structures in the fields of digital design and computer architecture. This assignment is based on Exercise 3.11.3 in the textbook.

Your task is to write two template classes for managing generic stacks and queues, based on the following requirements:

- Create and submit the files MyStack.h and MyQueue.h (remember that .cpp files are not allowed when defining templates). Ensure that the MyStack and MyQueue classes work properly with the `int`, `float`, and `char` datatypes.
- The maximum size of the stack or queue shall be hardcoded as 10.
- Each class should provide the following public member functions:
  - default constructor
  - peek: non-destructively returns the value on top of the stack or at the front of the queue. An exception handler should return an arbitrary value and print an error message if the container is empty.
  - pop: remove one element from container and return it
  - push: insert new element into container
  - clear: delete all entries from container
  - size: number of elements in container
  - full: returns true if the container is full
  - empty: returns true if the container is empty
- Overflow and underflow must be handled via exceptions.

The grader will compile your classes with a main.cpp file to grade functionality.

This main.cpp file:

```
#include <iostream>
#include <iostream>
#include "MyStack.h"
#include "MyQueue.h"

using namespace std;

int main()
{
   MyStack <int> s1;
   MyQueue <int> q1;
   int i;

   cout << "Filling stack and queue." << endl;

   for (i=0; i<11; i++)
   {
      s1.push(i);
      q1.push(i);
      cout << "Top of stack = " << s1.peek() << "\t" <<
              "Stack size = " << s1.size() << endl;
      cout << "Front of queue = " << q1.peek() << "\t" <<
              "Queue size = " << q1.size() << endl;
   }

   cout << "Emptying stack and queue." << endl;

   for (i=0; i<11; i++)
   {
      s1.pop();
      q1.pop();
      cout << "Top of stack = " << s1.peek() << "\t" <<
              "Stack size = " << s1.size() << endl;
      cout << "Front of queue = " << q1.peek() << "\t" <<
              "Queue size = " << q1.size() << endl;
   }

   return 0;
}
```

Should produce this output:

```
Filling stack and queue.
Top of stack = 0        Stack size = 1
Front of queue = 0      Queue size = 1
Top of stack = 1        Stack size = 2
Front of queue = 0      Queue size = 2
Top of stack = 2        Stack size = 3
Front of queue = 0      Queue size = 3
Top of stack = 3        Stack size = 4
Front of queue = 0      Queue size = 4
Top of stack = 4        Stack size = 5
Front of queue = 0      Queue size = 5
Top of stack = 5        Stack size = 6
Front of queue = 0      Queue size = 6
Top of stack = 6        Stack size = 7
Front of queue = 0      Queue size = 7
Top of stack = 7        Stack size = 8
Front of queue = 0      Queue size = 8
Top of stack = 8        Stack size = 9
Front of queue = 0      Queue size = 9
Top of stack = 9        Stack size = 10
Front of queue = 0      Queue size = 10
Exception:  Tried to push to a full stack!
Exception:  Tried to push to a full queue!
Top of stack = 9        Stack size = 10
Front of queue = 0      Queue size = 10
Emptying stack and queue.
Top of stack = 8        Stack size = 9
Front of queue = 1      Queue size = 9
```

```
Top of stack = 7        Stack size = 8
Front of queue = 2      Queue size = 8
Top of stack = 6        Stack size = 7
Front of queue = 3      Queue size = 7
Top of stack = 5        Stack size = 6
Front of queue = 4      Queue size = 6
Top of stack = 4        Stack size = 5
Front of queue = 5      Queue size = 5
Top of stack = 3        Stack size = 4
Front of queue = 6      Queue size = 4
Top of stack = 2        Stack size = 3
Front of queue = 7      Queue size = 3
Top of stack = 1        Stack size = 2
Front of queue = 8      Queue size = 2
Top of stack = 0        Stack size = 1
Front of queue = 9      Queue size = 1
Exception:  Tried to peek at an empty stack!
Top of stack = 0        Stack size = 0
Exception:  Tried to peek at an empty queue!
Front of queue = 0      Queue size = 0
Exception:  Tried to pop from an empty stack!
Exception:  Tried to pop from an empty queue!
Exception:  Tried to peek at an empty stack!
Top of stack = 0        Stack size = 0
Exception:  Tried to peek at an empty queue!
Front of queue = 0      Queue size = 0
```