

```

j@j-VirtualBox:~/cs/CPT_S-223/build$ make
Scanning dependencies of target MA1
[ 50%] Building CXX object CMakeFiles/MA1.dir/main.cpp.o
/home/j/c/c/CPT_S-223/main.cpp:12:1: error: new types may not be defined in a return type
12  class queue
    ^~~~~~

/home/j/c/c/CPT_S-223/main.cpp:12:1: note: (perhaps a semicolon is missing after the definition of 'queue')
/home/j/c/c/CPT_S-223/main.cpp:33:22: error: return type specification for constructor invalid
33  queue::queue(int size)
    ^~~~~~

/home/j/c/c/CPT_S-223/main.cpp: In member function 'void queue::dequeue()':
/home/j/c/c/CPT_S-223/main.cpp:59:24: error: invalid operands of types 'const char [10]' and 'int' to binary 'operator<<'
59      cout < "Removing " << arr[front] << '\n';
                        ^~   ^~~~~~
                        |      |
                        |      int
                        const char [10]

/home/j/c/c/CPT_S-223/main.cpp: In member function 'int queue::peek()':
/home/j/c/c/CPT_S-223/main.cpp:88:16: error: 'numeric_limits' was not declared in this scope
88      return numeric_limits<int>::min();
               ^~~~~~

/home/j/c/c/CPT_S-223/main.cpp:88:31: error: expected primary-expression before 'int'
88      return numeric_limits<int>::min();
                           ^~

/home/j/c/c/CPT_S-223/main.cpp:88:31: error: expected ';' before 'int'
88      return numeric_limits<int>::min();
                           ^~

/home/j/c/c/CPT_S-223/main.cpp:88:34: error: expected unqualified-id before '>' token
88      return numeric_limits<int>::min();
                           ^~

/home/j/c/c/CPT_S-223/main.cpp: In member function 'bool queue::isFull()':
/home/j/c/c/CPT_S-223/main.cpp:108:24: error: lvalue required as left operand of assignment
108     return (size()-1 = capacity);
                        ^~~~~~

make[2]: *** [CMakeFiles/MA1.dir/build.make:63: CMakeFiles/MA1.dir/main.cpp.o] Error 1
make[1]: *** [CMakeFiles/Makefile2:76: CMakeFiles/MA1.dir/all] Error 2
make: *** [Makefile:84: all] Error 2
j@j-VirtualBox:~/cs/CPT_S-223/build$

```

The screenshot shows the Visual Studio Code interface with the file `queue.h` open. The code implements a queue with the following methods:

- `enqueue(int item)`: Adds an item to the queue. It checks for overflow using `isFull()`. If full, it prints "OverFlow\nProgram Terminated\n". Otherwise, it increments the rear pointer, stores the item, and increments the count.
- `setCapacity(int x)`: Sets the capacity of the queue.
- `peek()`: Returns the front element of the queue, checking if the queue is empty.

The `main` function in `main.cpp` tests the queue by enqueuing the number 1 three times. The terminal output shows the successful execution of the program:

```

testing enqueue:
Inserting 1
Inserting 1
Inserting 1
Inserting 1

```