Alexander Bartella, Student ID: 400308868

Question 1

a) In c++ classes, members are designated by private by default. Members can be classified as private, protected, or public. Also, C structs can only have public members. On the other hand, C++ classes can contain member functions, while C structs cannot have a function as a member.

b) A copy constructor is strictly required for a class to function properly when an object has members that are dynamically allocated or passed by reference. If we copied without a copy constructor, the memory addresses of those members would be copied to the new object, not a new instance of said members. So, a copy constructor is needed to copy to a new object with unique members under these circumstances.

Question 2

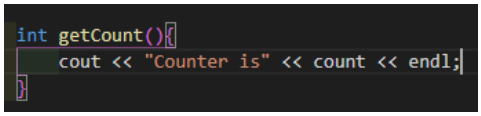
1) Line 6: constructor cannot have a return type. To fix, remove “void” keyword at the beginning of the line.

   void Counter(int c = 10) : count(c) { } ------> Counter(int c = 10) : count(c) { }

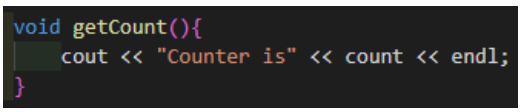
2) Line 8: destructors don’t require any input parameters. To fix, remove “int c” from inside the round brackets.

   ~Counter(int c){ } ------> ~Counter(){ }

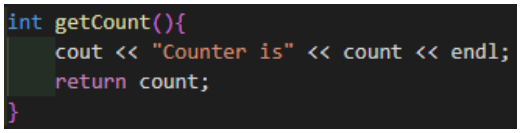
3) Line 14: member function “getCount()” has a return type of int but returns void. To fix, change return type from “int” at beginning of line 14 to “void”. Alternatively, return an integer at the end of the function declaration (after line 15).



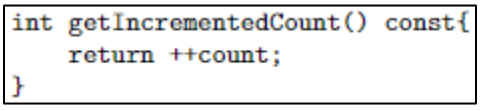
→



OR →



4) Line 10: member function “getIncrementedCount()” is a constant function, meaning it cannot change a member variable belonging to the “Counter” class. The issue is that the function is trying to modify the “count” member variable. To fix, remove “const” keyword from the end of line 10.



→