

### Compare and Contrast

*Instructions: Describe similarities and/or differences of each of the mentioned concepts. You may describe in plain English or C++ code.*

1. C vs C++. Is C++ a subset of ANSI C, a superset of ANSI C, or neither?

They are neither a superset nor a subset. Although C++ takes things from C we can't call it a superset.

2. Classes vs Structures. What is each, and what are the default access modifier? How do you change the default access control?

Classes are user-defined data type containing data members and functions. Private by default access modifier.

Structures create a single data type that can be used to group objects. The access modifier is public by default.

In order to change default access modifier the user would have to specify using public, private, or protected.

3. Encapsulation vs Inheritance. What can you do with encapsulation that you cannot with inheritance and vice versa?

Through encapsulation users can define the data however they want using identifiers and modifiers.

Inheritance allows a class to gain properties meaning it gain modifiers and identifier that could be shared with other classes but they can't be changed

4. Local vs Global scope. How would you define each type of scope, what are the limits in access, and what is the lifetime of each?

Local scope is only within the block it is declared in.

The Global scope is available everywhere within the script.

The life time of local is during the function it is in and Global variable is throughout the program.

5. Pointers vs References. What is the purpose of each element? Can you use them interchangeably?

`int* ptr = &i` Pointers holds a memory address of another variable.

`int &ref = i` References is a variable that has another name of another existing variable.

You can use them interchangeably

6. Linker vs Compiler. What are the functions/purpose of each? Does linking or compiling happen first? Can you use them in the other order?

Compiler generates object files and checks if your source code follows C++ rules.

Linkers use the object files created by compiler to create an executable

Compiling occurs first and cannot occur in the other order

7. What are the different types of polymorphism supported by C++?

There is run-time polymorphism which done by function overriding which is changing the definition of the member of the base class.

There is also compile-time polymorphism which is done through function overloading or operator overloading