Your name: Dennis Klimenkov
_____

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?

   Neither, while C++ took a lot of stuff from C, we can't call it a subset because not all things from C work in C++, and neither all things from C++ work in C.

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

A structure is a grouping of variables of various data types referenced by the same name, that serves as a template for creating an instance of the structure. While class is a user-defined object type with its own set of data members and member functions that can be accessed and used by creating a class instance.

By default, all the members of the structure are public and all members of the class are private.

To change the default access you need to implicitly set the access control.

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

   Encapsulation allows to separate data between public and private, that wouldn't allow usage of data outside of class. Inheritance allows to set a superclass that would share their public data with subclasses.

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?

int global;
int main(){ int local;}
print(global) // prints
print(local) // throws error

local scope is available only within a statement block while global scope is available everywhere

life time for local variable is the time while the statement block is running
life time of global variable is the length of program execution (while memory allocated)

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

Pointer holds a memory address to another variable while reference is an alias of the variable, that also access the variable by storing it's memory address.

Yes, you can use them interchangeably in order to get the variable that both pointer and reference hold.

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?

Linker connects all object for executable, in order for linker to link the object compiler needs to compile files into objects. Compiler must always go first in order for linker to link object and create an executable

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

Run-time Polymorphism and Compile-time polymorphism.
Also know function and/or operator overloading vs
function overriding