

Week 12

Intro to Computer Science - CSI 1430

Hello and welcome to the weekly resources for CSI 1430: Introduction to Computer Science! This week's resource will be covering the course material that will be covered in week 12 of the course.

Last week, we finished our introduction of classes. However, there is another way we can create objects in C++: using structs! This week, we will dive into that concept. **Just as a reminder:** These documents are intended for going over concepts that are discovered in class, as they are a concise and to the point review of the materials. But remember, these documents are not a replacement for lecture and they cannot cover everything, so make sure you get help on any specific concepts you struggle with as well as go over lecture notes, programs, and the textbook.

Reminder: If you have any questions about these study guides, group tutoring sessions, private 30-minute tutoring appointments, the Baylor Tutoring channel or any tutoring services we offer, please visit our website <https://baylor.edu/tutoring> or call our drop-in center during open business hours. **Monday-Thursday 9am - 8pm on class days (254) 710-4135.**

Keywords: *Structs*

TOPIC OF THE WEEK

Structs

It's a fairly common question to ask what the difference between a class and a struct are in C++, and which one should be used in each instance. **In C++, the only difference between classes and structs is that all member variables and functions of a struct are by default publicly accessible from outside of the struct, whereas all member variables and functions of a class are by default private.** For example, in the following snippet, *ExampleStruct* has three

member variables, and since none of these variables have a “private” access specifier preceding them, they are all automatically accessible from outside the struct.

```
struct ExampleStruct {  
    char a;  
    int b;  
    double c;  
};  
  
int main () {  
    ExampleStruct myStruct;  
  
    myStruct.b = 99;  
  
    return 0;  
}
```

As it stands, `myStruct's attributes`, such as `b`, `are able to be accessed directly`.

However, if for example, `ExampleStruct's 'b' attribute` was declared as:

```
private int b;
```

Then, `'myStruct.b = 99;'` would not compile because the `'b' attribute for the ExampleStruct class now has the private specifier`.

To contrast with a class, consider the following example:

```
class Person {  
    int age;  
    string name;  
}
```

This class currently has no access specifier and by default will not have its attributes able to be accessed. With this simple Person class, “age” and “name” are by default private and not accessible, unless you denote them as public (or you create getters/setters for them):

```
class Person {  
public:  
    int age;  
    string name;  
}
```

Now, if you were to instantiate a new *Person* object, you can freely access and change their age and name.

In conclusion, the choice between class and struct depends on the program’s need. The main benefit of a class is that its default private member variables allows for a more secure program, while a struct’s strength lies in its faster member access due to the default public member variables. Usually, if a program needs to store information that should not be easily changed, class is the way to go; if the program prioritizes accessibility of the stored data, structs may be better.

CHECK YOUR LEARNING

1. What type of access specifiers do struct attributes have by default?
2. What type of access specifiers do class attributes have by default?

THINGS STUDENTS MAY STRUGGLE WITH

1. There might be more differences between structs and classes depending on the programming language you use, so make sure to check specifics on concepts like that when transitioning from one programming language to another.
2. Helpful Resources:

- <https://www.geeksforgeeks.org/structure-vs-class-in-cpp/> :Very short 3 - 4 minute read about structs versus classes in C++.
 - https://www.youtube.com/watch?v=ABRP_5RYhqU : Short video on classes in C++, goes over access specifiers as well!
-

Thanks for checking out these weekly resources! Don't forget to check out our website for group tutoring times, video tutorials and lots of other course resources: <https://baylor.edu/tutoring>. The answers to the 'Check Your Learning' questions are below.

Answers:

1. Public
2. Private

Note: All tables were taken from the Baylor CS 1430 zyBook