**Overview of Basic Concepts in C++ - Chapter 4**

Table of Contents

[Structures 3](#_Toc66017940)

[Enumerations 4](#_Toc66017941)

Chapters 2 – 5 discuss some basic concepts about C++. All of the details in the book are not being covered here, only those that are new or important.

## Structures

Structures are defined in C++ as follow:

struct integers  
{  
 int a;  
 int b;  
 float c;  
}  
int main()  
{  
 integers int1;  
 int1.a = 2;  
 int1.b = 4;  
 int1.c = 5.2;  
}

C++

Notice that unlike in C, we do not have to declare that part is of type struct while creating a new member.

We can also copy all values from one structure to another like this:

part part2 = part1;

C++

This is because the structure part is essentially a variable type in C++ since it is an object.

In reality, structures in C++ can also hold functions, which they could not in C, and structures and classes can be used almost interchangeably, but it is not common practice to use structures for anything other than data.

## Enumerations

Enumerations are not generally used very often. They can define custom data types.

#include<iostream>  
using namespace std;  
  
enum days {Sun, Mon, Tue, Wed, Thu, Fri, Sat};  
int main()  
{  
 days day1 = Mon, day2 = Thu;  
 cout << day2 – day1;  
}

C++

An enum declaration defines the set of all permissible names of the type. These are called enumerators. We can then create a variable and set it to any of the enumerators. The variables cannot have values outside of the list. We can then perform mathematical operations on the variables, like the subtraction shown above, or comparisons like day1 < day2. In C, it was necessary to use the keyword enum while declaring a new variable, but this is not necessary in C++.

The list of enumerators are being treated as integers internally, with the first member being given the value of 0, which is why it is possible to perform arithmetic operations. However, a list like enum pets {Dog, Cat, Bird}, although giving arithmetic results, does not make much sense.

It is possible to assign different numerical values to an enumerator in the initial set by set it to whatever value is desired, such as Dog = 2. Also, although it is possible to make use of the numerical values of the enumerators, it is ill-advised. In fact, it is best to forget that they have integer values at all.

A small problem with enumerations is, in C++ I/O, the numerical values of the enumerators are used instead of the names.