

Lecture Transcripts

Basic Built-in Data Types

Hello and welcome to the course on Data Structures and Algorithms. This is the first session of this course in which we will look at some basic built-in data types in C++ programming language.

So if you think of data structures and algorithms, they are the building blocks of computer programs. There is a old book with title says DATA STRUCTURES + ALGORITHMS = PROGRAMS. A program essentially prescribed what is to be done with what data. And in a C++ program, it's made up of variables and functions. Where the variables correspond to the data structures in use and the functions correspond to the algorithms that are used.

So let us look at a simple example program in C++. This just reads two integers and outputs their sum. So if you see the declarations, we have declared two integer variables i and j. We read in their values and we output the sum of those two values. We need to include some library files in order to read in and output these integers. But for the moment we'll ignore those libraries, just concentrate on the declarations in the program.

So this program uses two variables and one function. The variables are declared to be integers, what that says is that these variables can store integer values. And we have only one function which is the main function which reads the two integer values and outputs their sum.

So if you look at the data structures used in this program the variables i and j correspond to the data structures use and they represent integer values. So this is a built-in a data structure int available as part of the programming language. Essentially, if you want to represent integers in a computer, you can use this built-in data type which is readily available in the programming language. A data type is also another name for a data structure, in a programming language we may call a variable to have a certain type and we can also call it as a data structure for representing values. And the data type of a variable defines what are its possible values and what are the operations that can be performed on it. So an int variable, for example, specifies that you can store integer values in that variable and you can perform arithmetic operations, you can perform comparisons and many other operations on variables of integer type.

Now, in this case, the algorithm used is just the main defined by the main function. This just uses again the built-in operation plus for adding int variables. You do not need to specify how to add two integer variables. Because the addition operation is already built-in for an int variable. This also uses functions for reading int and printing int variables. These functions are defined as part of the iostream library, but we will not be concern with them at the moment. But many commonly required data structures and algorithms are available as built-in types in C++ or as part of libraries. So in many of these cases, you do not need to build your own data structures or define your own types for representing data. But you can use what is available to represent whatever data you want.

So let's look at this program and some questions about this. So, will this program always correctly add two integers? So, under what condition is it guaranteed to work correctly? So, here there is an implicit assumption made when we use an int type to declare the variable. The int type has some restrictions on what integers can be represented. So, it is not guaranteed that this will always work correctly. Also, there are other built-in data structures that could have been used in this program. So, can you think of some possible structures? So, there are other ways of representing integers in a C++ program which could also have been used here. And now if you want to write a program which is guaranty to work for any integers, it should not matter how large the integer is or what are its value. Now, in this case, there is no built-in data structures that you can use for representing such integers. So, you will have to write your own data structures and write your own algorithms for performing the addition operation followed. So, the general idea in any program is we need to define what are the data structures needed, what are the algorithms needed and we define, if possible, use built-in data types. If not, we need to define our own data type.

So, we'll look at more examples of programs and use of data structures and algorithms in them in subsequent sessions.

Thank you.