



DOWNLOAD: <https://tinurli.com/2iky52>

[Download](#)

5.1 Introduction Data structures are a key part of the repertoire of programming techniques. This chapter presents a practical approach to the use of data structures in C and C++. It begins with a short introduction to the basic types of data structures and a comparison of their use in the C and C++ language. The chapter then presents, in a unified style, a number of structures with which programmers can work. Finally, a small number of small but extremely useful data structures are listed in which the programmer can look for efficient implementations. 5.2 Basic Types In C and C++, all data objects are expressed in terms of a structure called a "struct" (the "struct" suffix is omitted in C++). The structure contains a name, and a number of data members (which will be defined later in this chapter). A pointer to the structure's name is called a "pointer to struct". The type of a pointer to struct is not "struct" but "typedef" (typedef is a special way of creating names; see §1.4.5). A pointer to the first member of a structure (also a "pointer to struct") is called a "pointer to struct member". The data members are typed in terms of a built-in data type for the language; this type is called "basic type". The basic types are described in the standard text as a function of the type of the data members in a structure; so for example, the type of a member x of a structure S with data members y and z is described as the "declaration of S's data members", and the type of the data member z as "the type of S's data member z". The following three points should be kept in mind when writing programs in C and C++:

- (1) A pointer to struct is a special kind of pointer: it does not refer to a single data object but to a group of data objects. For example, the data members of a structure are accessed using pointer arithmetic. (2) In a pointer to struct, the name is accessed using a syntax called "pointer to member". This means that the left-hand side of the expression that is dereferenced must be a pointer to a structure; the structure must be specified (to the right of the pointer) by its name, and the name of the member to which the pointer points must be specified. For example, the pointer expression a[82157476af

[sureshaddin.xls](#)
[Fatal Twelve Mini Voice Drama Download Setup](#)
[Microsoft Windows 7 Ultimate Sp1 X64 Msdn Hun Torrent 58](#)