

---

# Before We Start

**Whom is the Book for?** *Beginners* who have just picked up C and want to try their hands on some data structures can use this book. This book could be used as a supplement to any undergraduate data structure course.

*Intermediate* programmers who have a grip over C and common data structures will come to know about other data structures like ‘tries’, ‘Hash Maps’ and how to design them using age-old C building blocks like arrays, structures, etc.

*Expert* programmers can use this book as a handy data structure reference. So this is a book for everyone who deals with data structures!!

*Happy Structuring!*

*Best of Luck ☺*

**Organization of this Book** Programming examples are the prime focus of this book. It attempts to teach its reader how to apply well-known data structures to solve problems in diverse areas/fields. There is an algorithm for approximate string matching that finds application in cellphones, DNA matching and statistical analysis for medicine grouping. Here, we will learn the algorithm as a topic with more inputs for its application to solve real-world problems, rather than beating around the theory of the algorithm.

**Compilers and OS** I have used Microsoft Visual Studio 6.0, Microsoft Visual Studio Express Edition and Turbo C(for some programs) to code the applications. But these codes can be compiled in Turbo C 3.5 or higher under DOS. Whenever any part of the code is compiler specific, that is clearly mentioned. My PC runs on *Microsoft Windows XP Professional*. You can use any Windows OS.

**What is a Data Structure?** A Data Structure is nothing but a container for data. As water is to a container, data are to data structures. In simple terms, it is a structure that could hold your data. Thus, the name *data structure* is justified.

**Why do we need One?** For simple applications like arithmetic manipulations, we can do away with user-defined data structures, but for rather serious applications where we need to store a lot of information like age, name, address, etc., we need to have things like ‘Data Structures’. Sometimes it may happen that any of the inbuilt data structures (although C has only one ‘array’) does not solve a particular need. Then we shall have to write our customized data structures using the available primitive data structures of the language.