FREE EBOOK

oliveboard

Data Structure, File Structure and Programming through ‘C’, ‘C++’, ‘Java’ & ‘Python’

Phase-II Study Notes

For NABARD Gr. A IT Officer Exam

Data Structure, File Structure and Programming through 'C', 'C++', 'Java' & 'Python'

Phase-II Study Notes for NABARD Gr. A IT Officer Exam

Data Structure

A data structure is a data organization, management, and storage format that is usually chosen for efficient access to data.

More precisely, a data structure is a collection of data values, the relationships among them, and the functions or operations that can be applied to the data, i.e., it is an algebraic structure about data.

Usage

- Data structures serve as the basis for abstract data types (ADT). The ADT defines the logical form of the data type.
- The data structure implements the physical form of the data type.
- Different types of data structures are suited to different kinds of applications, and some are highly specialized for specific tasks. For example, relational databases commonly use B-tree indexes for data retrieval, while compiler implementations usually use hash tables to look up identifiers.
- Data structures provide a means to efficiently manage large amounts of data for uses such as large databases and internet indexing services.
- Usually, efficient data structures are critical to designing efficient algorithms.
- Some formal design methods and programming languages emphasize data structures, rather than algorithms, as the key organizing factor in software design.
- Data structures can be used to organize the storage and retrieval of information stored in both main memory and secondary memory.

Implementation

- Data structures are generally based on the ability of a computer to fetch and store data at any place in its memory, specified by a pointer—a bit string, representing a memory address, that can be itself stored in memory and manipulated by the program.

- Thus, the array and record data structures are based on computing the addresses of data items with arithmetic operations, while the linked data structures are based on storing addresses of data items within the structure itself.

- The implementation of a data structure usually requires writing a set of procedures that create and manipulate instances of that structure.

- The efficiency of a data structure cannot be analysed separately from those operations.

- This observation motivates the theoretical concept of an abstract data type, a data structure that is defined indirectly by the operations that may be performed on it, and the mathematical properties of those operations (including their space and time cost).

There are numerous types of data structures, generally built upon simpler primitive data types. Here are some well-known examples:

- A byte is the smallest amount of data that many CPUs can copy from memory to a register or back in a single CPU instruction, therefore a byte stream is the most efficient way to run big data through a computer, hence Stream processing.

- An array is a number of elements in a specific order, typically all of the same type (depending on the language, individual elements may either all be forced to be the same type, or maybe of almost any type).

- Elements are accessed using an integer index to specify which element is required.

- +Typical implementations allocate contiguous memory words for the elements of arrays (but this is not always a necessity). Arrays may be fixed length or resizable.

- A linked list (also just called list) is a linear collection of data elements of any type, called nodes, where each node has itself a value, and points to the next node in the linked list.

- The principal advantage of a linked list over an array is that values can always be efficiently inserted and removed without relocating the rest of the list. Certain other operations, such as random access to a certain element, are however slower on lists than on arrays.

- A record (also called tuple or struct) is an aggregate data structure. A record is a value that contains other values, typically in fixed numbers and sequences and typically indexed by names. The elements of records are usually called fields or members.

- A union is a data structure that specifies which of a number of permitted primitive types may be stored in its instances, e.g., float or long integer. In contrast with a record, which could be defined to contain a float and an integer, whereas, in a union, there is only one value at a time. Enough space is allocated to contain the widest member data type.

- A tagged union (also called variant, variant record, discriminated union, or disjoint union) contains an additional field indicating its current type, for enhanced type safety.

- An object is a data structure that contains data fields, as a record does, as well as various methods which operate on the data contents.

- An object is an in-memory instance of a class from a taxonomy. In the context of object-oriented programming, records are known as plain old data structures to distinguish them from objects.

In addition, hashes, graphs, and binary trees are other commonly used data structures.

Language Support

- Most assembly languages and some low-level languages, such as BCPL (Basic Combined Programming Language), lack built-in support for data structures.

- On the other hand, many high-level programming languages, and some higher-level assembly languages, such as MASM, have a special syntax or other built-in support for certain data structures, such as records and arrays.

- For example, the C (a direct descendant of BCPL) and Pascal languages support structs and records, respectively, in addition to vectors (one-dimensional arrays) and multi-dimensional arrays.

- Most programming languages feature some sort of library mechanism that allows data structure implementations to be reused by different programs.

- Modern languages usually come with standard libraries that implement the most common data structures. Examples are the C++ Standard Template Library, the Java Collections Framework, and the Microsoft .NET Framework.

- Modern languages also generally support modular programming, the separation between the interface of a library module and its implementation. Some provide opaque data types that allow clients to hide implementation details.

- Object-oriented programming languages, such as C++, Java, and Smalltalk, typically use classes for this purpose.

- Many known data structures have concurrent versions which allow multiple computing threads to access a single concrete instance of a data structure simultaneously.

File System Structure

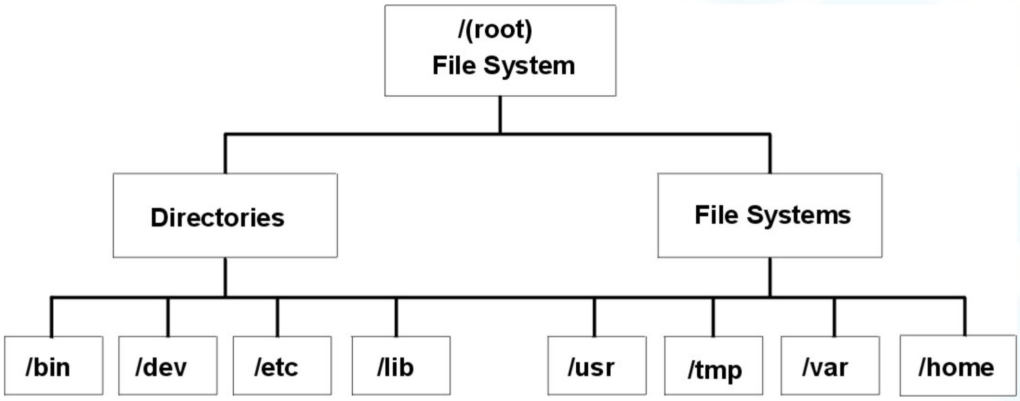
- It is important to understand the difference between a file system and a directory.

- A file system is a section of the hard disk that has been allocated to contain files. This section of the hard disk is accessed by mounting the file system over a directory.

- After the file system is mounted, it looks just like any other directory to the end user.

- However, because of the structural differences between the file systems and directories, the data within these entities can be managed separately.

- When the operating system is installed for the first time, it is loaded into a directory structure, as shown in the following illustration.



- The directories on the right (/usr, /tmp, /var, and /home) are all file systems so they have separate sections of the hard disk allocated for their use.

- These file systems are mounted automatically when the system is started, so the end user does not see the difference between these file systems and the directories listed on the left (/bin, /dev, /etc, and /lib).

- A programming language is composed of a set of instructions or codes that a computer can understand and execute.

Programming Languages

- A program is a set of instructions that tells a computer what to do in order to come up with a solution to a particular problem.

- Programs are written in a language that computers can understand, known as the programming language.

- Each programming language has its own syntax, which consists of a set of rules that dictate how words and symbols can be put together to form a program. There are hundreds of different programming languages, each with its own logic and syntax.

- Only a few of them are really popular, but a programmer can easily use a dozen or more languages during a career.

C, C++, and C#

- C is a general-purpose programming language with a long history.

- C is used for many different types of software, but it is particularly popular for system software, such as operating systems, device drivers and telecommunications applications.

- C is widely used because it runs very fast.

- It can also access a computer system's low-level functions; this means it is closer to the hardware than some other programming languages.

- C has become an official standard of the American National Standards Institute or ANSI. Many other programming languages borrow syntax from C.

- C++ is a high-level programming language that builds on its predecessor C by adding object-oriented features.

- C++ is very versatile and can be used for many different applications. C++ is also an official ANSI standard.

- C++ is used widely for applications that rely more heavily on a graphical user interface or GUI. For example, many utility programs and device drivers are written in C, while applications software with many user dialogs are written in C++.

- The name C++ represents the evolution of the C language because the two plus symbols represent the increment operator in C.

- C# is a programming language designed by Microsoft. It combines the functionality of C and C++ with Visual Basic.

- C# is used on many operating systems, not just Windows. C# is one of the languages used in the Microsoft .NET framework.

- While C++ supports both object-oriented programming and procedural programming, C# is strictly an object-oriented programming language. The name C# was inspired by musical notation where a sharp indicates that the note should be made a semitone higher in pitch.

- The sharp symbol also resembles the combination of four plus symbols to suggest that the language is an increment of C++.

- The Microsoft .NET framework is a programming infrastructure created by Microsoft. It is also referred to as Visual Studio .NET. It is used for building many different types of applications, such as desktop software applications and web services.

- The .NET framework makes it possible for different programming languages to work together. However, C# is the language specifically developed to create code for the .NET framework.

- Objective-C is also derived from C and has been adapted by Apple to develop applications for the Mac OS and iOS operating systems.

- Objective-C has therefore become very popular for mobile phone apps.

- The name Objective-C was chosen to make it clear that this was an object-oriented language with its origins in the C language.

Python Programming

- Free and open-source - You can freely use and distribute Python, even for commercial use.

- Easy to learn - Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like C++, Java, and C#.

- Portable - You can move Python programs from one platform to another and run them without any changes.

Python Features

- Python is easy to learn. Its syntax is easy, and the code is very readable.

- Python has a lot of applications. It's used for developing web applications, data science, rapid application development, and so on.

- Python allows you to write programs in fewer lines of code than most programming languages.

- The popularity of Python is growing rapidly. Now it's one of the most popular programming languages.

- Python, one of the most popular programming languages in the world, has created everything from Netflix's recommendation algorithm to the software that controls self-driving cars.

- Python is a general-purpose language, which means it's designed to be used in a range of applications, including data science, software and web development, automation, and generally getting stuff done.

What is Python?

- Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis.

- Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems.

- This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today.

What is Python used for?

- Python is commonly used for developing websites and software, task automation, data analysis, and data visualization.
- Since it's relatively easy to learn, Python has been adopted by many non-programmers such as accountants and scientists, for a variety of everyday tasks, like organizing finances.

Why is Python so Popular?

Python is popular for a number of reasons.

It has a simple syntax that mimics natural language, so it's easier to read and understand. This makes it quicker to build projects, and faster to improve on them.

- It's versatile.
- Python can be used for many different tasks, from web development to machine learning.
- It's beginner friendly, making it popular for entry-level coders.
- It's open source, which means it's free to use and distribute, even for commercial purposes.
- Python's archive of modules and libraries—bundles of code that third-party users have created to expand Python's capabilities—is vast and growing.
- Python has a large and active community that contributes to Python's pool of modules and libraries and acts as a helpful resource for other programmers.
- The vast support community means that if coders run into a stumbling block, finding a solution is relatively easy; somebody is bound to have run into the same problem before.

NABARD Grade A IT Officer Online Course

We have launched a comprehensive course for your all-round preparation for the NABARD Grade A IT officer post.

Cover the basics of **Computer/IT, ESI & ARD, and Decision-Making** through **Crisp Study Notes** and test your understanding through **Objective + Descriptive Mock Tests**.

Take a look at the course highlights given below:

oliveboard

Ace

NABARD Gr. A

2022

IT Officer Exam

Get a Comprehensive Course for Your All-Round Prep

Course Highlights

Study Notes

- IT/Computer
- Computer & Decision Making
- ESI & ARD
- Descr. English Paper
- Current Affairs

Mock Tests

- 10 Ph-I + 5 Ph-II Object. Tests
- 5 Descr. Tests with Evaluation

Enroll Now

Enroll Here

Connect with us on:

- Telegram
- Discuss Forum
- YouTube