Competitive Coding Interview Questions

190+ questions to tackle any C/C++ interview

Dr. Rydhm Beri

www.bpbonline.com

First Edition 2024

ISBN: 978-93-55517-616

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true to correct and the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.

Dedicated to

My Parents
Ms. Suman Beri
and
Mr. Rajesh Beri

About the Author

Dr. Rydhm Beri, presently working at Alliance University, Bengaluru, Karnataka holds a Doctorate in Computer Applications from Lovely Professional University, Punjab. She has completed her Master's degrees in Computer Science from Guru Nanak Dev University, Punjab, and in Computer Applications from Lovely Professional University. With seven and a half years of experience in academic and administrative roles at esteemed institutions, Dr. Beri has excelled as a Placement Coordinator, Program Leader, IQAC Coordinator, and Assistant Dean.

Over the past 5 years, Dr. Rydhm has been actively involved in implementing Outcome Based Education and Curriculum Development. As a certified Google educator, she has served as a Google Faculty Advisor for the Google Developer Student Club for two years and was selected as a Women Tech Maker Ambassador by Google, reflecting her dedication and proficiency.

Dr. Beri's primary interests encompass Programming, the Internet of Things, Cloud Computing, and Application Development. Her doctoral research focused on developing a Pregnancy E-Healthcare System using fog computing, facilitating remote monitoring of pregnant women's health, and providing recommendations in the absence of medical professionals. Her innovative contributions earned her the prestigious Confederation of Indian Industry (CII) Gold Award in 2022 for pioneering low-cost automation solutions. In 2021, she was honored with the Young Scientist Award from VD Good, recognizing her significant societal impact.

She serves as a reviewer for renowned publications including Springer, IEEE, Elsevier, and IGI Global, contributing to scholarly advancements. Dr. Beri has authored 8 books and published over 20 articles, presented insights at 9 prestigious conferences, and contributed more than 25 journal papers. Her work includes securing 10 patents for innovative ideas and obtaining 1 copyright for her creative contributions, highlighting her substantial achievements in academia and research.

About the Reviewers

- **Kevin Stinsen Harkins** is a demonstrator within the Programming Languages and Compilers Department at Eötvös Loránd University, teaching students fundamental concepts of C programming and low-level programming. He gained extensive industry experience at Morgan Stanley working on complex systems using Scala and Python, at IBM as an SAP developer, and Texas Instrument as a Field application engineer.

  He wrote his thesis on controlling a drone using brainwaves. Currently, he is conducting research on computer vision applied to 3D clouds using a LIDAR sensor, aiming to detect geometric primitives using digital morphology implemented in C, an older technique in image processing.

- **Mahima Singla** is a passionate Principal Software Design Engineer with extensive professional experience and passion for designing and implementing robust, scalable software solutions in the domain of Cloud Assessment, Cloud Governance, Cloud Cost Optimization, Application fitment for cloud, Cloud Operating Model, and cutting-edge technologies, particularly in the realms of Cloud computing, AWS, and Kubernetes in Go language.

  She is currently working in Precisely Software and is a part of the Studio Administrator Cloud and Customer Onboarding projects. She has played a pivotal role in architecting and developing solutions that harness the full potential of cloud platforms. Her proficiency in AWS extends to services like AWS services EC2, S3, and Lambda that has allowed her to create resilient and scalable applications that align with business objectives.

  Throughout her career, she has not only focused on technical excellence but also on driving innovation and fostering collaborative environments. As a Principal Software Engineer, she takes pride in leading teams to deliver high-quality solutions that meet and exceed client expectations.

Acknowledgement

First and foremost, I am deeply thankful to God for his unwavering grace and guidance, which has shaped every aspect of this book. I also extend my gratitude to BPB Publications for their expertise and support in bringing this book to life. The journey of revising this book has been long and demanding, and I am grateful for the valuable contributions and collaboration of reviewers, technical experts, and editors.

I would like to express a heartfelt thanks to my loving and supportive brother, Mr. Kartik Beri, and my mother for their endless encouragement and motivation. My gratitude also goes to my guru, Dr. Mithilesh Kumar Dubey, for his teachings and for standing by me through all the hardships of my life. I am thankful to all my teachers; without their guidance, I would not have reached this stage.

Lastly, and most importantly, I am grateful to all my critics. Your criticism has been a driving force, enabling me to improve and achieve my goals. Finally, I want to thank all the readers who have shown interest in my book and supported its realization. Your encouragement has been invaluable.

Preface

In today's rapidly evolving technological landscape, staying competitive in the field of software development requires a deep understanding of fundamental programming concepts and the ability to solve complex problems efficiently. Whether you are a recent graduate preparing for your first job interview or a seasoned professional looking to refresh your knowledge, this book aims to be your comprehensive guide to ace technical interviews in C, C++, Data Structures, and Database Management Systems (DBMS).

The journey to becoming a proficient software engineer is often paved with rigorous technical interviews that test your knowledge, problem-solving abilities, and coding skills. This book is designed to help you navigate this challenging path with confidence. It compiles a wide range of interview questions and answers, providing you with the insights and practice needed to excel.

Each chapter includes a series of questions that range from basic to advanced levels. The questions are designed to test various aspects of your knowledge and problem-solving skills. Detailed solutions and explanations are provided to help you understand the reasoning behind each answer.

Chapter 1: C Programming Core Concepts - presents the interview questions related to core C programming. This chapter includes questions that assess understanding of core concepts such as data types, pointers, memory management, and the use of arrays and structures. The questions cover the intricacies of pointer arithmetic, dynamic memory allocation (malloc, calloc, free), and string manipulation functions. Interviewers often probe knowledge of file I/O operations, preprocessor directives, and the compilation process. These problem-solving questions are included in the chapter which may involve writing and debugging code snippets, implementing algorithms, and explaining the time and space complexity of different solutions. The readers will understand how to use and implement various data structures, such as linked lists, stacks, and queues, which is also commonly evaluated.

Chapter 2: C Programming Complex Concepts - presents questions on control statements often focused on understanding the flow of execution through constructs like if-else, switch-case, loops (for, while, do-while), and their correct usage in different scenarios. Functions are essential for modularity, so the questions in this chapter may involve writing and optimizing them, understanding scope and lifetime of variables, and recursion.

Arrays, including multidimensional ones, are pivotal for handling collections of data, and questions may involve sorting, searching, and manipulating array elements. This chapter also covers questions related to Strings. Further, this chapter includes questions on pointer arithmetic, dynamic memory allocation, and pointer-to-pointer concepts. Structures and unions, and File handling related questions are also covered in this chapter.

Chapter 3: C++ Programming Core Concepts - presents the questions related to core concepts such as the syntax, data types, and basic input/output operations. Object-oriented programming (OOP) is a fundamental aspect of C++, so questions delve into the principles of encapsulation, inheritance, polymorphism, and abstraction, as well as the implementation and use of classes and objects. In interviews candidates might be asked to design class hierarchies, explain virtual functions, and discuss concepts like constructors, destructors, and operator overloading. Further, questions in this chapter also cover aspects like smart pointers (e.g., unique_ptr, shared_ptr), lambda expressions, the auto keyword for type inference, range-based for loops, and improvements in the standard library such as the addition of the std::thread for multithreading. Mastery of these topics demonstrates a candidate's ability to write modern, efficient, and maintainable C++ code.

Chapter 4: C++ Advanced Concepts - presents Standard Template Library (STL), which is pivotal for efficient and effective coding. Questions on STL containers explore various types like vectors, lists, deques, sets, and maps, focusing on their internal structures, performance trade-offs, and optimal use cases. STL algorithms, a powerful feature of the library, provide a wide range of functionalities such as sorting, searching, and modifying data, allowing developers to write concise and expressive code. Advanced STL questions delve into custom iterators, functors, lambda expressions, and the intricacies of template programming, including template specialization and variadic templates.

Chapter 5: Data Structures Core Concepts - In the field of computer science and software engineering, mastering the core concepts of data structures is fundamental for tackling technical interview questions and building efficient algorithms. This chapter delves into pivotal data structures such as arrays, which store elements in contiguous memory for fast access and manipulation. Linked lists extend this by using nodes with pointers, offering dynamic memory allocation and flexibility. This chapter further discussed questions related to Stacks, Trees, including binary trees and AVL trees, organizing data hierarchically, facilitating efficient searching, insertion, and deletion operations. Alongside, this chapter also covers Graphs, which consist of nodes and edges. Mastering these data structures equips engineers with the tools to design optimized solutions to complex problems in various domains.

Chapter 6: Database Management System - In the realm of database management and software engineering, understanding core data structure concepts is crucial for navigating technical interview questions and building robust database solutions. This chapter explores essential topics such as ER data modeling, which helps in designing databases by defining entities and their relationships. The chapter further has questions related to relational model and normalization techniques to ensure efficient data storage and retrieval by reducing redundancy and maintaining data integrity. Other topics covered in this chapter include, Transaction and concurrency control mechanisms, and Structured Query Language (SQL) that serves as the standard language for managing relational databases, allowing for efficient querying, data manipulation, and schema definition. Proficiency in these concepts empowers engineers to design scalable, secure, and efficient database systems that meet real-world application demands.

Code Bundle and Coloured Images

Please follow the link to download the
Code Bundle and the ***Coloured Images*** of the book:

https://rebrand.ly/kx9hct9

The code bundle for the book is also hosted on GitHub at
https://github.com/bpbpublications/Competitive-Coding-Interview-Questions.
In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

https://discord.bpbonline.com

Table of Contents

Chapter 1
C Programming Core Concepts

Introduction

C programming language is the first thing that every programmer or beginner learn at the beginning of his career. C programming language plays a vital role for getting selected in your dream organization. Based on the technical interview requirement, this chapter focusing on the majority of the technical questions asked from the core concepts of C programming.

Structure

In this chapter, we will discuss the following topics:

- C programming language
- Compilers and interpreters
- Header files in C programming
- Directives in C programming
- Variables in C programming
- Constants in C programming

- Datatypes in C programming
- Operators in C programming

Objectives

By the end of this chapter, you will be able to understand local and global scope of variables and their lifetimes. Further, you can distinguish between, and handle both binary and text files.

Section 1: C programming language

Question 1: **What is C programming?**

Answer: C is a general-purpose, procedural programming language that was originally developed at *Bell Labs* in the early 1970s by Dennis Ritchie. C is a high-level language that allows developers to write programs for a variety of applications, including operating systems, embedded systems, and various other computer applications.

C is a compiled language, which means that the source code is written in a human-readable format and then translated into machine-readable code by a compiler. This process results in an executable file that can be run on a computer.

C is known for its efficiency, portability, and low-level programming capabilities, which make it an ideal choice for system programming and other tasks that require low-level access to computer hardware. It is also widely used for developing software applications in a variety of fields, including finance, gaming, and scientific computing.

Question 2: **Some of the key features of the C language include its simple syntax, powerful memory management capabilities, and ability to directly manipulate bits and bytes of data. C has influenced the development of many other programming languages, including *C++*, *Java*, and *Python*. Who developed the C programming language and for what purpose was it developed?**

Answer: The C programming language was developed by Dennis Ritchie at *Bell Labs* in the early 1970s. Ritchie developed C as a successor to the B programming language, which was used for systems programming on the **Multics** operating system.

C was designed to be a portable, high-level language and provides low-level access to memory and language constructs that could be used for a wide range of tasks, including systems programming, application development,

and scientific computing. Ritchie wanted to create a language that was efficient, flexible, and easy to learn and use, while still providing low-level access to computer hardware.

One of the main goals of the development of C was to create a language that could be used to develop the Unix operating system, which was also being developed at *Bell Labs* at the time. C proved to be well-suited for this task, and Unix was eventually rewritten entirely in C.

C quickly gained popularity among programmers, and its simplicity and efficiency led to its widespread adoption for a variety of programming tasks. The language has since become one of the most widely used programming languages in the world, and it continues to be used for a wide range of applications, from operating systems to scientific simulations to web development.

Question 3: What are the features of the C programming language?

Answer: Here are some of the key features of the C programming language:

- **Procedural programming:** C is a procedural programming language, which means that it allows developers to structure their code into functions and procedures.
- **Low-level programming capabilities:** C allows developers to directly manipulate memory and perform low-level operations, making it well-suited for system programming.
- **Portable:** C code can be compiled on different platforms, making it a portable language.
- **Efficient:** C is known for its efficiency in terms of memory usage and execution speed.
- **Static typing:** C is a statically typed language, which means that variables must be declared with a specific data type before they can be used.
- **Simple syntax**: The syntax of C is relatively simple, making it easy to read and write.
- **Standard library:** C provides a standard library of functions that can be used in a wide range of applications.
- **Pointers:** C allows developers to use pointers, which are variables that store memory addresses, giving them direct access to memory.

- **Structured programming:** C supports structured programming, which involves dividing code into reusable blocks of code, making it easier to manage and debug.
- **Recursion**: C supports recursion, which is the ability of a function to call itself.
- **Bit manipulation**: C allows developers to manipulate individual bits of data using bitwise operators.
- **Dynamic memory allocation**: C provides functions for dynamic memory allocation, which allows developers to allocate memory during runtime.
- **Extensibility:** C can be extended through the use of libraries and interfaces to other languages.
- **Preprocessor directives:** C provides preprocessor directives, which allow developers to modify the source code before compilation.
- **User-defined data types:** C allows developers to define their own data types using structures and unions.
- **Modular programming:** C supports modular programming, which involves dividing code into modules that can be developed and tested independently.
- **Command line interface:** C programs can be executed from the command line, making it easy to automate tasks.
- **Multi-platform support:** C can be compiled and executed on a wide range of platforms, including Windows, Linux, and macOS.
- **Large developer community:** C has a large developer community, with many resources and tools available for developers.
- **Influence on other programming languages:** C has had a significant influence on the development of other programming languages, including C++, *Java*, and *Python*.

Question 4: What is the basic structure of C program?

Answer: C is structural programming language, and follows set of lines to be written strictly in defined structure. The typical structure of C program includes:

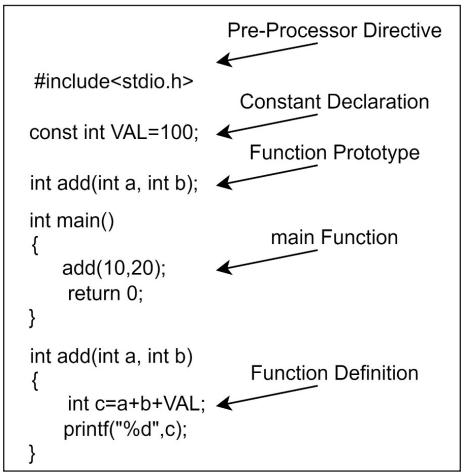

Figure 1.1: *Program Structure*

- **Preprocessor directives**: This section includes preprocessor directives that start with a # symbol. These directives are processed by the preprocessor before actual compilation. Common preprocessor directives include including header files (**`#include`**) and defining constants & macros (**`#define`**).

  For example

  ```
  #include<stdio.h>
  #define PI 3.1427
  ```

- **Global variables and constants:** This section contains global variable declarations and constants that will be used throughout the program.

  For example:

  ```
  int a=20;
  const int MAX=100;
  ```

- **Function prototype:** This section declares the prototypes of functions that are defined later in the program. Function prototypes tell the compiler about the functions' names, return types, and parameters.

  For example:

  ```
  int addNumbers(int a, int b);
  ```

- **Main function:** Every C program must have a main function from which the execution of the program begins.