Top 10 Computer Programmer Interview Questions and Answers [Updated 2024]

Description

If you’re preparing for a computer programmer interview, it’s vital to familiarize yourself with the types of questions you might face. From understanding your coding expertise to your problem-solving skills, this guide lists the top questions you may encounter and offers insights on how to respond to them.

Computer Programmer Interview Questions

Can you explain the concept of Object-Oriented Programming (OOP)?

How to Answer

Try to cover the basic principles of OOP: Encapsulation, Inheritance, Polymorphism, and Abstraction. It’s also beneficial to provide real-world examples or scenarios where each principle can be applied.

Sample Answer

Object-Oriented Programming (OOP) is a programming paradigm that is based on the concept of ‘objects’, which can contain data and code. Data in the form of fields, also known as attributes, and code in the form of procedures, also known as methods. In OOP, the computer programs are designed using the concept of objects that interact with real-world entities. The four basic principles of OOP are Encapsulation, Inheritance, Polymorphism, and Abstraction. Encapsulation is the mechanism that binds code and data together and keeps them safe from outside interference or misuse. Inheritance is a way of creating a new class using properties of an existing class. Polymorphism allows one interface to be used for a general class of actions. And abstraction is a mechanism to represent the essential features without including background details.

Can you describe the difference between ‘==’ and ‘===’ in JavaScript?

How to Answer

When responding to this question, you should focus on the technical differences between the two operators and provide examples for better understanding. You should mention the concepts of type coercion and strict equality in JavaScript.

Sample Answer

‘==’ in JavaScript is called the loose equality operator. It compares two values for equality, after converting both to a common type. On the other hand, ‘===’ is the strict equality operator that checks if the two values are equal in type and in value, without doing any type coercion. For instance, if we compare 2==’2? it will return true because ‘==’ converts the string ‘2’ to a number before comparison. However, 2===’2? will return false because ‘===’ doesn’t do any type conversion, and number 2 and string ‘2’ are not the same type.

Can you explain how a merge sort algorithm works?

How to Answer

You should first explain what a merge sort is, which is a divide and conquer algorithm that divides a collection into smaller ones, sort them and then merge them. You should also explain the process of how it works. It’s also a good idea to give an example to demonstrate your explanation.

Sample Answer

Merge sort is a type of sorting algorithm that follows a divide and conquer approach. It involves dividing the unsorted list into N sublists, where each sublist contains one element, which is considered sorted. The process of merging takes place, where the smaller sublists are merged into a new list, maintaining the sorted order. The process continues until we get a sorted list of N elements. For example, if we have an array of [38, 27, 43, 3, 9, 82, 10], it would first be divided into subarrays until each array only contains one element, then they would be merged back together in a sorted order.

MASTERING THE INTERVIEW COMPUTER PROGRAMMER

Your Ultimate Guide to Success 🚀

Mock Interview Pro

Can you explain the difference between a stack and a queue in computer science?

How to Answer

In the answer, the candidate should first define both the stack and the queue data structures briefly. Then, they should explain the main differences between the two, such as the order of elements removal (LIFO for stack and FIFO for queue). They can also mention some use cases for both data structures to demonstrate deeper understanding.

Sample Answer

A stack is a data structure that follows the Last-In-First-Out (LIFO) principle. That means the last element added to the stack will be the first one to be removed. Stacks are often used in recursion implementation, parsing, and memory management. On the other hand, a queue is a data structure that follows the First-In-First-Out (FIFO) principle. So, the first element added to the queue will be the first one to be removed. Queues are commonly used in scenarios where we want to maintain the order of elements, like in CPU scheduling, Disk Scheduling, etc.

Can you describe how recursion works in programming?

How to Answer

Start by defining what recursion is, then explain the concept of a base case and a recursive case. Provide an example to illustrate your explanation, it could be a simple one like factorial or Fibonacci sequence. Remember to mention the pros and cons of using recursion.

Sample Answer

Recursion in programming refers to a function that calls itself in order to solve a problem. This method is based on the principle of solving a complex problem by breaking it down into smaller, more manageable problems. A recursive function typically has two parts: a base case, which defines the solution to a simple instance of the problem, and a recursive case that expresses the solution in terms of a call to the same function on a smaller instance of the problem. For example, the calculation of a factorial is a common use case for recursion. The base case would be that the factorial of 0 is 1, and the recursive case would define the factorial of any other number n as n times the factorial of (n-1). While recursion can make the code cleaner and easier to understand, it can also lead to performance issues if not used carefully, as each recursive call adds a layer to the system’s call stack.

? Click to practice this and numerous other questions with expert guidance

Can you explain the concept of polymorphism in programming?

How to Answer

To properly answer this question, you should first define polymorphism and then illustrate it with an example. It’s also important to explain why polymorphism is useful and how it enhances flexibility and reusability in code.

Sample Answer

Polymorphism is one of the four fundamental principles of Object-Oriented Programming (OOP). It allows routines to use variables of different types at different times. An operator or function can behave differently in different contexts. For instance, let’s take an example of a ‘draw’ method in a ‘Shape’ class. This method can be overridden by each derived class like ‘Circle’, ‘Rectangle’, ‘Triangle’ etc. Each ‘draw’ method will have a different implementation according to the specific shape. So, when we call the ‘draw’ method, it will execute the appropriate method related to the specific object. This is polymorphism. Polymorphism is essential in OOP as it enhances the flexibility and reusability of the code, allowing the same name to be used for methods that perform a similar but not identical task.

Can you describe the concept of inheritance in Object-Oriented Programming?

How to Answer

To answer this question, first define what inheritance is in the context of programming. Next, explain how it works in OOP, and discuss its purpose or benefits in software development. You can also provide an example to better illustrate the concept.

Sample Answer

Inheritance in Object-Oriented Programming is a feature that allows a class (child class) to acquire the properties and methods of another class (parent class). This is an essential feature in OOP as it promotes code reusability and a form of organization where properties and methods are defined once and used multiple times. For example, if we have a class ‘Vehicle’ with properties like ‘color’ and ‘speed’, and methods like ‘accelerate’ and ‘brake’, we can create another class ‘Car’ that inherits from ‘Vehicle’. This means the ‘Car’ class will automatically have ‘color’ and ‘speed’ properties, as well as ‘accelerate’ and ‘brake’ methods, without us having to define them again.

Can you describe how garbage collection works in Java and how it can impact performance?

How to Answer

The candidate should start by explaining the concept of garbage collection in Java, i.e., how it automatically frees up memory that is no longer needed by removing objects that are not being used by the application anymore. They should then discuss the different types of garbage collectors, such as the Serial Collector, Parallel Collector, and G1 Collector, and how they work. Finally, they should discuss how garbage collection can impact performance, particularly if it is not managed correctly.

Sample Answer

Garbage collection in Java is a process that automatically reclaims memory that is no longer needed. This is done by removing objects that are not being used by the application anymore. There are different types of garbage collectors in Java, such as the Serial Collector, which is designed for single-threaded environments, the Parallel Collector, which is designed for multi-threaded environments, and the G1 Collector, which is designed for systems with large amounts of memory. If not managed correctly, garbage collection can have a negative impact on performance. For example, if the garbage collector runs too frequently, it can slow down the application as it uses CPU cycles. On the other hand, if the garbage collector does not run frequently enough, it can lead to memory problems as unused objects are not removed in a timely manner.

Can you explain how exception handling works in Python?

How to Answer

The candidate should start by defining what an exception is in Python, and then explain how the try/except block is used to catch exceptions. They should also mention the optional else and finally clauses, and explain what they’re used for. Finally, they should talk about how to raise exceptions using the raise keyword.

Sample Answer

Exception handling in Python is done using the try/except statement. When the code within a try block encounters an error during its execution, the flow of control is passed to a corresponding except block, which contains code to handle the error. There are also optional else and finally clauses. The else block is executed if the try block does not raise any exceptions, and it is typically used for code that must be executed if the try block is successful. The finally block is always executed, regardless of whether an exception was raised or not, and it is typically used for cleanup code. Exceptions can also be manually raised using the raise keyword, followed by the name of the exception to raise.

Can you explain the concept of multithreading and its benefits in programming?

How to Answer
When answering this question, start by defining multithreading. Then, discuss its benefits such as improved performance and responsiveness, and how it allows for simultaneous execution of two or more parts of a program. Also, mention its use in background tasks and UI updates. Lastly, talk about the challenges that come with multithreading, such as synchronization and shared resources issues.

Sample Answer
Multithreading is an execution model that allows multiple threads to exist within the context of a single process, sharing the process resources but able to execute independently. The primary benefit of multithreading is that it increases the efficiency and performance of CPU by utilizing the idle time. It allows for simultaneous execution of two or more parts of a program for maximum utilization of CPU. Also, it’s commonly used for background tasks, UI updates, and tasks that could block the main thread. However, multithreading comes with its own challenges, such as dealing with synchronization and shared resource issues.

? Boost your confidence — practice this and countless questions with our help today

Download Computer Programmer Interview Questions in PDF

To make your preparation even more convenient, we’ve compiled all these top Computer Programmer interview questions and answers into a handy PDF.

Click the button below to download the PDF and have easy access to these essential questions anytime, anywhere:

Click here to download the PDF

Computer Programmer Job Title Summary

Job Description	A Computer Programmer is responsible for creating and designingsoftware and mobile apps, debugging software, and maintainingoperating systems. They use computer languages such as C++,Python, Java, and many more. They may also customize software to aclient’s needs and troubleshoot any problems that may arise.
Skills	Proficiency in several programming languages, Strong problem-solving skills, Attention to detail, Excellent communication skills, Understanding of software development principles, Ability to work in a team, Knowledge of database management
Industry	Information Technology, Software Services, Finance, Healthcare, Government, Education
Experience Level	Entry to Senior Level. Entry-level programmers may need at least a year of experience, while senior-level programmers may need several years of experience.
Education Requirements	A Bachelor’s degree in computer science, information systems, or a related field is typically required. Some positions may require a Master’s degree.
Work Environment	Computer Programmers typically work in an office setting. They often work full-time and may need to work extra hours to meet deadlines. Much of their work is done on a computer.
Salary Range	The salary range for a Computer Programmer in the United States can range from $50,000 to over $130,000 per year, depending on experience, skills, and the industry.
Career Path	Many Computer Programmers start as Junior Programmers and then move on to positions such as Senior Programmer, Project Manager, or Software Developer. Some may choose to specialize in a specific area of programming.
Popular Companies	Microsoft, Google, Apple, IBM, Amazon

MASTERING THE INTERVIEW COMPUTER PROGRAMMER

Your Ultimate Guide to Success 🚀

Mock Interview Pro

Land Your Dream Computer Programmer Job: Your Ultimate Interview Guide

Expert Strategies to Stand Out and Get Hired

? **Conquer Interview Nerves**: Master techniques designed for Computer Programmer professionals.
? **Showcase Your Expertise**: Learn how to highlight your unique skills
?? **Communicate with Confidence**: Build genuine connections with interviewers.
? **Ace Every Stage**: From tough interview questions to salary negotiations—we've got you covered.

Don't Leave Your Dream Job to Chance!
Get Instant Access