**Interview questions**

**C++**

**Basic**

* **Question:** What is the difference between a class and a struct in C++?  
  **Answer:** The only technical difference is the default access specifier. Members of a struct are public by default, while members of a class are private by default. In practice, they are functionally identical.
* **Question:** Name the four main categories of data types in C++.  
  **Answer:** The four main categories of built-in data types are:
  + **Integer types** (int, short, long, long long)
  + **Floating-point types** (float, double)
  + **Character types** (char, wchar\_t)
  + **Boolean type** (bool)
* **Question:** Can you compile a C++ program without a main() function, and if so, can you run it?  
  **Answer:** Yes, a C++ source file can be compiled into an object file without a main() function. However, it cannot be linked into a standalone executable because the linker requires main() as the program's entry point. Therefore, it cannot be run on its own.

**Intermediate**

* **Question:** Explain the difference between call-by-value and call-by-reference, and describe a scenario where you would use each.  
  **Answer:**
  + **Call-by-value:** A copy of the argument's value is passed to the function. Modifications to the parameter inside the function do not affect the original argument. Use this when the function needs to work with data without altering the source variable.
  + **Call-by-reference:** An alias (reference) to the original argument is passed. Modifications to the parameter inside the function directly affect the original argument. Use this to modify the caller's variable or to avoid the performance overhead of copying large objects.
* **Question:** What is polymorphism, and how does compile-time polymorphism differ from runtime polymorphism?  
  **Answer:** Polymorphism allows objects of different classes to be treated as objects of a common base class.
  + **Compile-time polymorphism** is resolved during compilation and is achieved through function overloading and operator overloading.
  + **Runtime polymorphism** is resolved during execution and is achieved through virtual functions and function overriding in a class hierarchy.
* **Question:** Define a copy constructor and explain the circumstances under which it is invoked.  
  **Answer:** A copy constructor is a special constructor that creates a new object as a copy of an existing object. It is invoked when:
  + An object is initialized from another object of the same class.
  + An object is passed to a function by value.
  + An object is returned from a function by value.

**Advanced**

* **Question:** Explain the concept of "perfect forwarding" and how it is implemented using variadic templates and std::forward.  
  **Answer:** Perfect forwarding is a C++ feature that allows a function to pass its arguments to another function while preserving their original value categories (lvalue or rvalue). It is implemented using forwarding references (also known as universal references) in a template function and std::forward to cast the argument to its original category. This avoids unnecessary copies and enables move semantics through generic wrapper functions.
* **Question:** What is an abstract class, and how do you create one? Provide a simple example demonstrating its use with derived classes.  
  **Answer:** An abstract class is a class that cannot be instantiated on its own and is designed to be a base class for other classes. You create one by declaring at least one pure virtual function (a virtual function assigned to 0). Derived classes must implement all pure virtual functions to become concrete (instantiable) classes.
* **Question:** Describe what a move constructor is and how its purpose and performance implications differ from a copy constructor.  
  **Answer:** A move constructor is a constructor that transfers ownership of resources (like heap-allocated memory) from a temporary (rvalue) object to a new object. Its purpose is to avoid expensive deep copies. Unlike a copy constructor, which allocates new resources and copies data, a move constructor performs a cheap "pilfering" of pointers and other internal data, leaving the source object in a valid but unspecified state. This significantly improves performance when dealing with temporary objects.

**Python**

**Basic**

* **Question:** What is the difference between a global and a local variable scope in Python?  
  **Answer:** A **local variable** is defined inside a function and can only be accessed within that function. A **global variable** is defined outside of any function and can be accessed throughout the program, both inside and outside functions (using the global keyword to modify it within a function).
* **Question:** Explain the purpose of the \_\_init\_\_() method within a Python class.  
  **Answer:** The \_\_init\_\_() method is a constructor that is automatically called when a new instance (object) of a class is created. Its primary purpose is to initialize the object's attributes with the values passed during instantiation.
* **Question:** What is the difference between a Python list and a tuple, and in what situation would you choose one over the other?  
  **Answer:** A list is mutable, meaning its elements can be changed, added, or removed. A tuple is immutable; once created, its elements cannot be changed. Use a list for collections of items that need to be modified, and a tuple for collections of items that should remain constant, such as dictionary keys or coordinates.

**Intermediate**

* **Question:** What are list comprehensions, and how can they be used as a more concise alternative to a for loop? Provide an example.  
  **Answer:** List comprehensions provide a concise syntax for creating a new list based on the values of an existing iterable. They replace a for loop, an optional if condition, and the list append() method with a single line. For example, squares = [x\*\*2 for x in range(10)] is the list comprehension equivalent of a for loop that calculates the squares of numbers from 0 to 9.
* **Question:** Explain the concept of a decorator in Python and provide a simple use case.  
  **Answer:** A decorator is a function that takes another function as an argument, adds some functionality, and returns the modified function without altering the original function's source code. A common use case is adding logging or timing capabilities to a function to measure its execution time.
* **Question:** How would you read data from a CSV file and process it into a list of dictionaries, where each dictionary represents a row?  
  **Answer:** Use Python's built-in csv module. Specifically, open the file and use csv.DictReader to iterate over the rows. DictReader automatically uses the first row as keys for the dictionaries it creates for each subsequent row.

**Advanced**

* **Question:** Describe what metaclasses are in Python and explain a practical scenario where using one would be beneficial.  
  **Answer:** A metaclass is a "class of a class"—it defines the rules by which a class is created. By default, the metaclass is type. A practical use case is in creating an ORM (Object-Relational Mapper), where a metaclass can automatically generate database query methods on a model class based on its defined fields, without the developer needing to write them explicitly.
* **Question:** Explain the difference between concurrency and parallelism in Python, and discuss the standard libraries used to achieve them.  
  **Answer:**
  + **Concurrency:** Multiple tasks make progress over overlapping time periods but don't necessarily run simultaneously. This is useful for I/O-bound tasks. It is achieved using the asyncio and threading libraries.
  + **Parallelism:** Multiple tasks run at the exact same time on different CPU cores. This is for CPU-bound tasks. It is achieved using the multiprocessing library.
* **Question:** How does Python's memory management work, specifically regarding reference counting and the garbage collection cycle detector?  
  **Answer:** Python primarily uses **reference counting**. Every object has a count of how many references point to it. When the count drops to zero, the object's memory is deallocated. To handle reference cycles (e.g., A refers to B, and B refers to A), Python has a supplemental **cycle detector** garbage collector that periodically finds and deallocates these unreachable cycles.

**Database (MySQL)**

**Basic**

* **Question:** What is the functional difference between the DELETE, TRUNCATE, and DROP statements in MySQL?  
  **Answer:**
  + DELETE: A DML command that removes rows from a table one by one and can be rolled back. It can use a WHERE clause.
  + TRUNCATE: A DDL command that quickly removes all rows from a table by deallocating the data pages. It cannot be rolled back and resets auto-increment counters.
  + DROP: A DDL command that completely removes the entire table, including its structure and data.
* **Question:** How do you use the DISTINCT keyword in a SELECT statement, and what is its purpose?  
  **Answer:** The DISTINCT keyword is used directly after SELECT to return only unique (different) values from one or more specified columns. Its purpose is to eliminate duplicate rows from the query result set.
* **Question:** What are primary keys and foreign keys, and what role do they play in maintaining relational database integrity?  
  **Answer:**
  + **Primary Key:** A constraint that uniquely identifies each record in a table. It must contain unique values and cannot contain NULL values.
  + **Foreign Key:** A key used to link two tables together. It is a field in one table that refers to the PRIMARY KEY in another table. They enforce referential integrity, ensuring relationships between tables remain valid.

**Intermediate**

* **Question:** Explain the difference between INNER JOIN, LEFT JOIN, and RIGHT JOIN, and provide a scenario for using each one.  
  **Answer:**
  + **INNER JOIN**: Returns only the rows where the join condition is met in both tables. Use it to find matching records, like customers who have placed orders.
  + **LEFT JOIN**: Returns all rows from the left table and the matched rows from the right table. If there is no match, the result is NULL from the right side. Use it to find all customers, including those who have not placed any orders.
  + **RIGHT JOIN**: Returns all rows from the right table and the matched rows from the left table. It is the functional inverse of a LEFT JOIN.
* **Question:** What is database normalization? Briefly explain the first three normal forms (1NF, 2NF, 3NF).  
  **Answer:** Database normalization is the process of organizing columns and tables in a relational database to minimize data redundancy and improve data integrity.
  + **1NF (First Normal Form):** Ensures that table cells hold single, atomic values and each record is unique.
  + **2NF (Second Normal Form):** Requires the table to be in 1NF and all non-key attributes to be fully functionally dependent on the entire primary key.
  + **3NF (Third Normal Form):** Requires the table to be in 2NF and all attributes to be dependent only on the primary key, not on other non-key attributes (no transitive dependencies).
* **Question:** What is an index in a database? Explain why indexes are used to improve query performance and discuss their potential downsides.  
  **Answer:** An index is a data structure (like a B-tree) that improves the speed of data retrieval operations on a database table. It works like an index in a book, allowing the database to find rows with specific column values much faster than scanning the entire table. The downside is that indexes consume storage space and slow down data modification operations (INSERT, UPDATE, DELETE) because the index must also be updated.

**Advanced**

* **Question:** Explain the concept of a database transaction and the meaning of the ACID properties (Atomicity, Consistency, Isolation, Durability).  
  **Answer:** A transaction is a single logical unit of work that consists of one or more SQL operations. ACID properties guarantee transaction validity even in the event of errors or power failures.
  + **Atomicity:** All operations within the transaction are completed successfully, or none of them are. It's an "all or nothing" principle.
  + **Consistency:** A transaction brings the database from one valid state to another, preserving all predefined rules.
  + **Isolation:** Concurrent transactions produce the same result as if they were executed sequentially. One transaction's intermediate state is not visible to others.
  + **Durability:** Once a transaction has been committed, its changes are permanent and will survive any subsequent system failure.
* **Question:** Describe a situation where you would implement a stored procedure in MySQL. What are its primary advantages and disadvantages?  
  **Answer:** A stored procedure would be used to encapsulate a complex business logic that is executed frequently, such as creating a new user account, which involves multiple INSERT and UPDATE statements across several tables.
  + **Advantages:** Reduced network traffic, centralized business logic, improved security, and better performance due to pre-compilation.
  + **Disadvantages:** Logic is tied to the database vendor, can be difficult to debug, and requires specialized DBA skills to manage.
* **Question:** What is database replication? Explain a common replication topology, such as master-slave replication, and its purpose.  
  **Answer:** Database replication is the process of copying and maintaining database objects in multiple databases to ensure data is available where it is needed. In **master-slave replication**, one database server (the master) logs all its changes, which are then read and applied by one or more other servers (the slaves). Its purpose is to provide read scalability (directing read queries to slaves), high availability, and disaster recovery.

**Web Designing**

**Basic**

* **Question:** Explain the fundamental roles of HTML, CSS, and JavaScript in creating a website.  
  **Answer:**
  + **HTML (HyperText Markup Language):** Provides the core structure and content of a web page.
  + **CSS (Cascading Style Sheets):** Defines the visual presentation, layout, and styling of the HTML content.
  + **JavaScript:** Adds interactivity, dynamic behavior, and complex logic to the web page.
* **Question:** What does the term "above the fold" mean in web design, and why is the content in this area considered so important?  
  **Answer:** "Above the fold" refers to the portion of a webpage that is visible to a user without scrolling. This content is critical because it creates the first impression and must immediately engage the user and communicate the site's purpose.
* **Question:** What is responsive web design, and why is it essential for a modern website?  
  **Answer:** Responsive web design is an approach that makes web pages render well on a variety of devices and window or screen sizes. It uses fluid grids, flexible images, and CSS media queries to adapt the layout. It is essential because it ensures a good user experience across desktops, tablets, and mobile phones, which now account for a majority of web traffic.

**Intermediate**

* **Question:** Describe your approach to handling cross-browser compatibility issues during the design and development process.  
  **Answer:** The approach involves using standard-compliant code, employing CSS resets or normalizers to create a consistent baseline, feature detection for modern APIs, and progressive enhancement. For testing, I use tools like BrowserStack or manual testing on major browsers (Chrome, Firefox, Safari, Edge) to identify and fix inconsistencies, often with vendor prefixes or polyfills for older browsers.
* **Question:** What are some key best practices for optimizing a website's loading performance?  
  **Answer:** Key best practices include: minimizing HTTP requests by combining files, compressing images and using modern formats like WebP, minifying CSS and JavaScript files, leveraging browser caching, using a Content Delivery Network (CDN), and deferring the loading of non-critical JavaScript.
* **Question:** Explain the concept of "progressive enhancement" as a web design strategy.  
  **Answer:** Progressive enhancement is a strategy where you build a baseline of essential content and functionality that works for all browsers. Then, you add layers of more advanced presentation and functionality (like complex CSS or JavaScript) that are available only to more capable browsers. This ensures universal access while providing a richer experience for users with modern technology.

**Advanced**

* **Question:** When starting a large-scale project, what CSS organization methodology (such as BEM or SMACSS) would you use, and why?  
  **Answer:** For a large-scale project, I would use the BEM (Block, Element, Modifier) methodology. Its strict, name-spacing convention (block\_\_element--modifier) creates highly modular, reusable, and self-documenting components. This structure minimizes the risk of selector conflicts and makes the CSS codebase more predictable and maintainable for a large team.
* **Question:** What are the Web Content Accessibility Guidelines (WCAG), and what steps do you take to ensure your designs are accessible to users with disabilities?  
  **Answer:** WCAG is a set of international standards providing recommendations for making web content more accessible. To ensure compliance, I use semantic HTML (e.g., <nav>, <main>), provide alt text for all meaningful images, ensure sufficient color contrast, implement keyboard navigability, use ARIA roles where necessary, and test with screen readers.
* **Question:** What are the most critical questions you would ask a client at the beginning of a project to understand their business goals and technical requirements?  
  **Answer:** Critical questions include:
  + What is the primary goal of this website (e.g., lead generation, e-commerce, information)?
  + Who is the target audience?
  + What specific actions do you want users to take?
  + What are the key performance indicators (KPIs) for success?
  + Are there any existing brand guidelines, technical constraints, or third-party integrations to consider?

**Graphic Designing**

**Basic**

* **Question:** What are the seven fundamental elements of graphic design?  
  **Answer:** The seven fundamental elements are Line, Shape, Color, Texture, Type (Typography), Space (White Space), and Form.
* **Question:** Explain the difference between raster and vector graphics, and provide an example of when to use each format.  
  **Answer:**
  + **Raster graphics** are composed of pixels. They are resolution-dependent and lose quality when scaled up. Use them for detailed images like photographs.
  + **Vector graphics** are composed of mathematical paths. They are resolution-independent and can be scaled to any size without losing quality. Use them for logos, icons, and illustrations.
* **Question:** What is typography, and how does it contribute to the effectiveness of a design?  
  **Answer:** Typography is the art and technique of arranging type to make written language legible, readable, and appealing when displayed. It contributes to a design's effectiveness by establishing a visual hierarchy, setting a tone or mood, and enhancing readability to clearly communicate the message.

**Intermediate**

* **Question:** Describe key principles of design composition, such as the golden ratio, proximity, and the use of white space.  
  **Answer:**
  + **Golden Ratio:** A mathematical ratio (approximately 1:1.618) used to create organic, aesthetically pleasing compositions.
  + **Proximity:** The principle of grouping related elements together to create a visual unit, which helps organize information and reduce clutter.
  + **White Space:** The negative or empty space around design elements. It is used to create focus, improve legibility, and provide a clean, uncluttered layout.
* **Question:** Briefly explain color theory and the practical difference between the RGB and CMYK color models.  
  **Answer:** Color theory is the body of practical guidance on color mixing and the visual effects of specific color combinations.
  + **RGB (Red, Green, Blue)** is an additive color model used for digital screens. Colors are created by adding light.
  + **CMYK (Cyan, Magenta, Yellow, Key/Black)** is a subtractive color model used for print. Colors are created by subtracting brightness from white paper as ink is applied.
* **Question:** In your design work, how do you use the lasso tool and its variations, such as the polygonal and magnetic lasso?  
  **Answer:**
  + **Lasso Tool:** Used for making freehand selections of irregular shapes.
  + **Polygonal Lasso Tool:** Used for making selections with straight edges by creating a series of connected line segments.
  + **Magnetic Lasso Tool:** Used to automatically "snap" the selection outline to the edges of high-contrast objects, making it efficient for selecting well-defined items from a background.

**Advanced**

* **Question:** Walk me through your typical design process, from receiving an initial brief to delivering the final assets.  
  **Answer:** My process is:
  1. **Discovery & Briefing:** Understand client goals, target audience, and project requirements.
  2. **Research & Mood Boarding:** Analyze competitors and gather visual inspiration.
  3. **Sketching & Ideation:** Develop low-fidelity concepts and wireframes.
  4. **Digital Design:** Create high-fidelity mockups based on the chosen concept.
  5. **Feedback & Iteration:** Present the design to the client and refine it based on their feedback.
  6. **Finalization & Delivery:** Prepare and export all final assets in the required formats.
* **Question:** How do you approach a project when a client provides negative feedback or requests major revisions that you believe are detrimental to the design?  
  **Answer:** I first listen to understand the root cause of their feedback, linking it back to the project's original goals. I then educate the client by explaining the design principles behind my choices and how they support those goals. I offer to present alternative solutions that meet their concerns while preserving the design's integrity, seeking a collaborative compromise rather than a direct confrontation.
* **Question:** Describe a design project that you found particularly challenging. What was the core problem, how did you approach it, and what was the final outcome?  
  **Answer:** A challenging project involved creating a brand identity for a tech startup with a highly abstract service. The core problem was visually communicating a complex, intangible concept to a non-technical audience. I approached this by focusing on the *benefit* of the service rather than the *process*. Through extensive research and collaborative workshops with the client, we developed a simple, metaphor-based visual system. The final outcome was a clean, minimalist logo and brand guide that effectively conveyed trust and simplicity, which was well-received by their target market.