

27 Lecture - CS304

Important Subjective

1. What is specialization in C++?

Answer: Specialization in C++ is a mechanism that allows programmers to define a different implementation of a template or function for a specific set of arguments.

Why is specialization useful in C++?

Answer: Specialization is useful in C++ when the default behavior of a template or function is not suitable for a particular data type or value.

What are the restrictions of specialization in C++?

Answer: Some of the restrictions of specialization in C++ include: you cannot partially specialize function templates, you cannot specialize function templates for built-in types, and specialization can lead to code duplication and maintenance issues.

How does specialization help in C++?

Answer: Specialization helps in C++ by allowing programmers to define a different implementation of a template or function for a specific set of arguments.

What is partial specialization in C++?

Answer: Partial specialization in C++ is a technique that allows programmers to define a specialized implementation of a template or function for a subset of the arguments.

Can you partially specialize class templates in C++?

Answer: Yes, you can partially specialize class templates in C++.

What is the syntax for specialization in C++?

Answer: The syntax for specialization in C++ is: `template <>`
`function_name<>(){/implementation/}`

What is explicit specialization in C++?

Answer: Explicit specialization in C++ is a technique that allows programmers to provide a separate implementation for a specific set of template arguments.

Can you specialize a non-template function in C++?

Answer: No, you cannot specialize a non-template function in C++.

What is a specialization hierarchy in C++?

Answer: A specialization hierarchy in C++ is a set of specialized implementations of a template or function that are ordered from the most general to the most specific.