

35 Lecture - CS304

Important Subjective

1. What are member templates in C++?

Answer: Member templates are templates that are defined inside a class or struct.

What is the advantage of using member templates?

Answer: Member templates provide increased reusability and adaptability of code, reduced code complexity, and improved code performance.

Can member templates access the data members and methods of the class they are defined in?

Answer: Yes, member templates can access the data members and methods of the class they are defined in.

What is the syntax for defining a member template?

Answer: The syntax for defining a member template is: `template<typename T> void MyClass<T>::myFunction(T arg) { //function body }`

Can member templates be specialized for specific data types?

Answer: Yes, member templates can be specialized for specific data types.

What is the difference between a member function template and a regular member function?

Answer: A member function template can work with any data type, whereas a regular member function can only work with specific data types.

What is the purpose of a member function template?

Answer: The purpose of a member function template is to provide a generic member function that can work with any data type.

Can member templates be used to provide generic constructors?

Answer: Yes, member templates can be used to provide generic constructors.

What is template specialization?

Answer: Template specialization is the process of defining a specialized version of a template for a specific data type.

What is the advantage of using a member function template over a regular member function?

Answer: The advantage of using a member function template is increased reusability and adaptability of code, reduced code complexity, and improved code performance.