

# 39 Lecture - CS304

## Important Subjective

1. **What is a template in C++?**

Answer: A template is a feature of C++ that allows for the creation of generic functions and classes that can work with different data types.

**What is a static member in C++?**

Answer: A static member is a member of a class that is shared by all instances of the class and can be accessed without creating an object.

**What is the purpose of using templates in C++?**

Answer: The purpose of using templates in C++ is to create generic functions and classes that can work with different data types, improving code reusability, flexibility, and efficiency.

**How is a static member variable initialized in C++?**

Answer: A static member variable is initialized outside the class definition, typically in the source file, using the scope resolution operator and the class name.

**Can a template class have static members?**

Answer: Yes, a template class can have static members that are shared by all instances of the class.

**What is template specialization in C++?**

Answer: Template specialization is a feature of C++ that allows for the creation of specialized versions of a template function or class for a specific data type.

**Can a static member function access non-static members of a class?**

Answer: No, a static member function cannot access non-static members of a class.

**Can a template function be overloaded in C++?**

Answer: Yes, a template function can be overloaded with different argument types.

**What is the syntax for declaring a static member function in C++?**

Answer: The syntax for declaring a static member function in C++ is to use the "static" keyword before the function name in the class definition.

**How can a template function be defined outside the class definition in C++?**

Answer: A template function can be defined outside the class definition by specifying the template parameter list and using the "template" keyword followed by the function signature.