

19 Lecture - CS304

Important Subjective

1. What is the purpose of the stream insertion operator in C++?

The stream insertion operator `<<` is used to output data to a stream, such as `cout`.

How do you overload the stream insertion operator for a user-defined class?

To overload the stream insertion operator `<<` for a user-defined class, you need to define a non-member function that takes an output stream `std::ostream&` and the class object as arguments.

Can the stream insertion operator be used to output multiple variables in a single statement?

Yes, the stream insertion operator can be chained to output multiple variables in a single statement, for example `std::cout << a << " " << b << std::endl;`

How can you control the formatting of output using the stream insertion operator?

You can use various manipulators such as `std::setw` and `std::setprecision` to control the formatting of output using the stream insertion operator.

What is the difference between the `<<` operator and the `put` method of the output stream class?

The `<<` operator is used to output data to a stream in a formatted way, whereas the `put` method is used to output individual characters to a stream.

How can you overload the stream insertion operator for a class that has private member variables?

You can make the stream insertion operator a friend function of the class, which allows it to access private member variables.

Can you use the stream insertion operator to output objects of built-in types like `int` or `double`?

Yes, the stream insertion operator can be used to output objects of built-in types like `int` or `double`.

What is the return type of the stream insertion operator?

The return type of the stream insertion operator is a reference to the output stream.

Can the stream insertion operator be used with input streams?

No, the stream insertion operator `<<` is used only for output streams. The stream extraction operator `>>` is used for input streams.

How do you handle errors that occur while using the stream insertion operator?

You can use the `std::ios::failbit` flag to check for errors that occur while using the stream insertion operator, and handle the errors appropriately.