

16 Lecture - CS304

Important Subjective

1. What is operator overloading in C++?

Answer: Operator overloading is a feature in C++ that allows operators such as +, -, *, /, and others to be redefined for user-defined types or classes.

What is the syntax for overloading the assignment operator in C++?

Answer: The syntax for overloading the assignment operator is: `ClassName& operator=(const ClassName& obj)`.

What is the difference between a unary operator and a binary operator?

Answer: A unary operator operates on a single operand, while a binary operator operates on two operands.

What is the significance of the friend keyword in operator overloading?

Answer: The friend keyword allows a non-member function to access the private and protected members of a class, which is useful for overloading certain operators.

What is the purpose of overloading the stream insertion and extraction operators in C++?

Answer: The purpose of overloading the stream insertion and extraction operators is to allow objects of user-defined classes to be formatted and read from input/output streams in the same way as built-in types.

Can the scope resolution operator (::) be overloaded in C++?

Answer: No, the scope resolution operator cannot be overloaded in C++.

What is the difference between the prefix and postfix increment operators in C++?

Answer: The prefix increment operator (`++x`) increments the operand before returning its value, while the postfix increment operator (`x++`) increments the operand after returning its value.

What is the syntax for overloading the addition operator in C++?

Answer: The syntax for overloading the addition operator is: `ClassName operator+(const ClassName& obj)`.

What is the difference between a member function and a non-member function for operator overloading?

Answer: A member function is a function that is a member of a class and operates on the object itself, while a non-member function is not a member of the class and operates on one or more objects of the class.

Can the conditional operator (?:) be overloaded in C++?

Answer: No, the conditional operator cannot be overloaded in C++.