

33 Lecture - CS304

Important Subjective

1. **What are multiple type arguments, and why are they useful in generic programming?**

Answer: Multiple type arguments refer to the ability to define multiple data types for use with a generic class or function. They are useful in generic programming because they allow for increased flexibility and reusability of code.

How are multiple type arguments defined in Java, and what is the default type argument?

Answer: Multiple type arguments are defined using the < > syntax in Java, and the default type argument is Object.

How are multiple type arguments defined in C++, and how many can be defined for a generic class?

Answer: Multiple type arguments are defined using the template < > syntax in C++, and any number can be defined for a generic class.

What is type erasure in the context of multiple type arguments?

Answer: Type erasure is the process of removing the generic type information from a generic class or method during compilation, allowing for backward compatibility with older code that was not designed to use generics.

Can multiple type arguments be used with functions in C++?

Answer: Yes, multiple type arguments can be used with functions in C++.

How do multiple type arguments improve code maintainability?

Answer: By reducing the need for duplicate code, multiple type arguments can improve code maintainability by making it easier to modify and update code.

How are multiple type arguments used in object-oriented programming?

Answer: Multiple type arguments are used in object-oriented programming to create reusable code that can be used with different data types.

What is the difference between single type arguments and multiple type arguments?

Answer: Single type arguments can only be used with one data type, while multiple type arguments can be used with multiple data types.

What is the syntax for defining multiple type arguments in Java?

Answer: The syntax for defining multiple type arguments in Java is < type1, type2, ... >.

What are some common use cases for multiple type arguments in generic programming?

Answer: Common use cases for multiple type arguments include the creation of generic algorithms, data structures, and collections that can be used with different data types.