

36 Lecture - CS304

Important Mcqs

1. What is member templates revisited in C++?

- a) A way to create specialized versions of member function templates
- b) A technique to simplify the syntax of member function templates
- c) A type of member function template that only works with specific data types
- d) A way to access private data members of a class using member templates

Answer: b) A technique to simplify the syntax of member function templates

What is the benefit of using member templates revisited?

- a) Reduced code duplication
- b) Improved performance
- c) Simplified syntax
- d) All of the above

Answer: d) All of the above

What is template argument deduction?

- a) The process of providing explicit template arguments to a function template
- b) The process of inferring template arguments from function arguments
- c) The process of specializing a template for a specific data type
- d) The process of defining a template inside a class

Answer: b) The process of inferring template arguments from function arguments

What is the syntax for using member templates revisited?

- a) `auto func(args...)`
- b) `template auto func(args...)`
- c) `template<typename T> auto Class::func(T arg)`
- d) `template<typename T> auto Class<T>::func(args...)`

Answer: d) `template<typename T> auto Class<T>::func(args...)`

Can member templates revisited be used with constructors?

- a) Yes
- b) No

Answer: a) Yes

What is the difference between regular member function templates and member function templates revisited?

- a) Member function templates revisited use template argument deduction
- b) Member function templates revisited can only be used with specific data types
- c) Regular member function templates have a simpler syntax
- d) Regular member function templates cannot be specialized

Answer: a) Member function templates revisited use template argument deduction

What is the purpose of template argument deduction in member templates revisited?

- a) To simplify the syntax of member function templates

- b) To reduce code duplication
- c) To allow for specialization of member function templates
- d) To infer the template arguments from the function arguments

Answer: d) To infer the template arguments from the function arguments

Can member templates revisited be used with non-static member functions?

- a) Yes
- b) No

Answer: a) Yes

What is the advantage of using member templates revisited over regular member function templates?

- a) Reduced code duplication
- b) Improved performance
- c) More concise and readable code
- d) All of the above

Answer: d) All of the above

What is the disadvantage of using member templates revisited?

- a) Limited support for certain compilers
- b) Increased complexity
- c) Slower compile times
- d) None of the above

Answer: d) None of the above