

23 Lecture - CS201

Important Subjective

- 1. What is the purpose of the pre-processor in C programming?**
Answer: The pre-processor performs pre-processing tasks such as handling pre-processor directives and including header files before the code is compiled.
- 2. What is a macro in C programming?**
Answer: A macro is a pre-processor directive that defines a text replacement that is expanded by the pre-processor.
- 3. How is a macro defined in C programming?**
Answer: A macro is defined using the `#define` directive followed by the macro name and its replacement text.
- 4. What is the purpose of the `#include` directive in C programming?**
Answer: The `#include` directive is used to include header files that contain function prototypes, constant definitions, and other declarations needed in the source code.
- 5. What is the purpose of the `#ifdef` directive in C programming?**
Answer: The `#ifdef` directive is used to include or exclude blocks of code depending on whether a certain macro has been defined.
- 6. How is a macro undefined in C programming?**
Answer: A macro is undefined using the `#undef` directive followed by the macro name.
- 7. What is conditional compilation in C programming?**
Answer: Conditional compilation is the process of including or excluding blocks of code based on certain conditions such as the target platform or the compiler being used.
- 8. What is the purpose of the `#pragma` directive in C programming?**
Answer: The `#pragma` directive is used to specify implementation-specific behavior or provide hints to the compiler.
- 9. What are the potential risks of using pre-processor directives excessively in C programming?**
Answer: Overuse of pre-processor directives can lead to code that is hard to read, maintain, and debug. It can also make the code more error-prone.
- 10. Can pre-processor directives be used in languages other than C?**
Answer: Yes, many programming languages have pre-processor directives, including C++, Objective-C, and Fortran.