5 Lecture - CS410

Important Subjective

Question 1:

What are preprocessor directives in C/C++?

Answer: Preprocessor directives are commands that instruct the compiler to perform specific actions before the actual compilation process. They begin with a '#' symbol and are used to manipulate the source code before it is compiled.

Question 2:

What is the purpose of the #include directive in C/C++?

Answer: The #include directive is used to include the content of another file (usually header files) into the current source file. It allows the use of functions, constants, and other declarations from the included file.

Question 3:

How do you define a macro using the #define directive in C/C++?

Answer: The #define directive is used to define macros in C/C++. For example, to define a macro for a constant, you can use:

```c
#define PI 3.14159
```

Question 4:

Explain the use of the #ifdef and #ifndef directives in C/C++.

Answer: #ifdef and #ifndef are used for conditional compilation. #ifdef checks if a macro is defined, and #ifndef checks if a macro is not defined. They are often used to control whether a specific block of code should be included in the final program.

Question 5:

How do you concatenate two tokens into a single token using the ## operator in C/C++?

Answer: The ## operator is used for token pasting or concatenation. For example:

```c
#define CONCAT(x, y) x ## y

int result = CONCAT(10, 20); // This will be replaced as int result = 1020;
```

Question 6:

What is the purpose of the #pragma directive in C/C++?

Answer: The #pragma directive provides implementation-specific instructions to the compiler. It is used for non-standard compiler-specific operations or settings.

Question 7:

How do you undefine a previously defined macro in C/C++?

Answer: The #undef directive is used to undefine a previously defined macro. For example:

```c
#define MAX_VALUE 100

#undef MAX_VALUE
```

Question 8:

What does the #error directive do in C/C++?

Answer: The #error directive is used to generate a compilation error message with a custom message. It is often used to communicate specific requirements or constraints during the compilation process.

Question 9:

Explain the purpose of the #if, #elif, and #else directives in C/C++.

Answer: #if, #elif, and #else are used for conditional compilation based on preprocessor macros. They allow different blocks of code to be included or excluded from the final program depending on specific conditions.

Question 10:

What is the difference between #include <filename> and #include "filename" in C/C++?

Answer: The #include <filename> is used to include standard library header files, while #include "filename" is used to include user-defined header files. The preprocessor searches for the standard library headers in system directories and user-defined headers in the current directory first before searching in system directories.