SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

COURSE NAME : 23ITT101- PROBLEM SOLVING & C PROGRAMMING

I YEAR /I SEMESTER

Unit V – STRUCTURE AND UNION

Topic : Preprocessor Directives

Topics Covered

- **Preprocessor Directives**
- Types of Preprocessor Directives

Preprocessor Directives

- Preprocessor directives in C are instructions to the compiler to preprocess the code before actual compilation begins. These directives start with the # symbol and are not part of the program code itself but provide important configurations and instructions for compilation.

Types of Preprocessor Directives

1. Macro Definition: #define
2. Conditional Compilation: #if, #ifdef, #ifndef, #else, #elif, #endif
3. File Inclusion: #include
4. Line Control: #line
5. Error Directive: #error
6. Pragma Directive: #pragma

1. Macro Definition (#define)

```
//Used to define constants or macros
#include <stdio.h>
#define PI 3.14159
#define SQUARE(x) ((x) * (x))  // Macro with argument

int main() {
    printf("PI: %.2f\n", PI);
    printf("Square of 5: %d\n", SQUARE(5));
    return 0;
}
```

Output:

```
PI: 3.14
Square of 5: 25
```

2. Conditional Compilation (#ifdef, #ifndef, etc.)

```
//Used to include or exclude parts of the code based on conditions
#include <stdio.h>
#define DEBUG 1  // Enable debug mode
int main() {
#ifdef DEBUG
    printf("Debugging is enabled.\n");
#endif
#ifndef RELEASE
    printf("Release mode is disabled.\n");
#endif
    return 0;
}
```

Output:

Debugging is enabled.
Release mode is disabled.

3. File Inclusion (#include)

//Used to include header files or user-defined files

//Angle brackets (<>): Search in standard directories.

//Quotes (""): Search in the current directory first, then standard directories.

```
#include <stdio.h>   // Standard header file
#include "myheader.h" // User-defined header file
int main() {
    printf("Hello, World!\n");
    return 0;
}
```

Output:

Hello, World!

4. Error Directive (#error)

```
//Used to generate a compilation error with a custom message
#include <stdio.h>
#ifndef VERSION
#error "VERSION is not defined!"
#endif
int main() {
    printf("This will not compile if VERSION is undefined.\n");
    return 0;
}
```

Output:

This will not compile if VERSION is undefined

5. Pragma Directive (#pragma)

```
//Used to issue special commands to the compiler (compiler-specific)
#include <stdio.h>
#pragma message("This is a compiler message.")
// Compiler shows this message

int main() {
    printf("Hello, World!\n");
    return 0;
}
```

Output:

This will not compile if VERSION is undefined

Common Use Cases

- Avoiding Multiple File Inclusions (Header Guards):

```
#ifndef HEADER_FILE
#define HEADER_FILE

// Code inside header file

#endif
```

Common Use Cases

- Platform-Specific Code:

```
#ifdef _WIN32
    printf("This code is for Windows.\n");
#else
    printf("This code is for other platforms.\n");
#endif
```

Debugging Mode with Conditional Compilation

```
#include <stdio.h>
// Macro to enable or disable debug mode
#define DEBUG 1
int main() {
    int a = 5, b = 3, sum;
    sum = a + b;
```

```
#ifdef DEBUG
    // Code included only if DEBUG is defined
    printf("[DEBUG] a: %d, b: %d, sum: %d\n", a, b, sum);
#endif
    // Regular program output
    printf("Sum is: %d\n", sum);
    return 0;
}
```

Explanation

- #define DEBUG 1: Enables debug mode by defining the DEBUG macro.
- #ifdef DEBUG: Includes the debug-related code (printf for internal values) only if DEBUG is defined.
- If you remove or comment out the #define DEBUG line, the debug messages will not be included in the final output.

Output

- If DEBUG is defined

[DEBUG] a: 5, b: 3, sum: 8

Sum is: 8

- If DEBUG is not defined

Sum is: 8


Thank
you