C Preprocessors

The preprocessor is a phase which occurs prior to compilation of a program.

The preprocessor has two main uses:

* it allows external files to be included and
* it allows macros to be defined.

Preprocessor directives can appear anywhere in a source file but they apply only to the remainder of the source file.

Preprocessor commands are distinguished by the pound symbol #.

# #include Preprocessor Directive:

The #include preprocessor directive causes a copy of specified file to be included in place of the directive. The two forms of #include directive are

1. #include<filename>

The preprocessor searches the standard library headers as #include<stdio.h>

1. #include ”filename.h”

The preprocessor searches the same directory as the file to be compiled for the file to be included, used in programs with multiple source files.

# #define Preprocessor Directive: Symbolic Constants

The #define directive creates symbolic constants. The define directive format is:

#define identifier replacement-text

For example: #define PI 3.14159

## Example

#include <stdio.h>

#include <stdlib.h>

#define PI 3.14159

double circleArea(double);

double circleCircumference(double);

int main(int argc, char \*argv[]) {

double r;

printf("Please enter the circle radius: ");

scanf("%lf",&r);

printf("The circle area is: %lf \n",circleArea(r));

printf("The circle area is: %lf \n",circleCircumference(r));

return 0;

}

double circleArea(double radius){

return radius \* radius \* PI;

}

double circleCircumference(double radius){

return 2 \* PI \* radius;

}

# 3.#define Preprocessor Directive: Macros

A macro is an identifier in a #define preprocessor directive. The macro identifier is replaced in the program with the replacement-text before the program is compiled

**#define MAX(A, B) ( (A) > (B) ? (A) : (B))**

**#define ABS(x) ((x) < 0)? -(x): (x))**

## Example:

**#include <stdio.h>**

**#include <stdlib.h>**

**#define MAX(A, B) ( (A) > (B) ? (A) : (B))**

**#define ABS(x) ((x) > 0 ? (x) : -(x))**

**int main(int argc, char \*argv[]) {**

**int a, b;**

**printf("Please enter the first number: ");**

**scanf("%d",&a);**

**printf("Please enter the second number: ");**

**scanf("%d",&b);**

**printf("The greater number is: %d\n", MAX(a,b));**

**printf("The absolute value of %d is: %d \n",a,ABS(a));**

**printf("The absolute value of %d is: %d \n",b,ABS(b));**

**return 0;**

**}**

# 4.Conditional compilation

Using conditional statements to control the preprocessing.

**#include <stdio.h>**

**#include <stdlib.h>**

**#define VERSION 1**

**int main(int argc, char \*argv[]) {**

**#if VERSION == 0**

**printf("Program version is 0");**

**#elif VERSION == 1**

**printf("Program version is 1");**

**#else**

**printf("Program version is 1");**

**#endif**

**#if !defined(MYAGE)**

**#define MYAGE 22**

**#endif**

**return 0;**

**}**