

Conditional compilation

- **Conditionals in the C pre-processor**

- **Conditional clause of the C pre-processor:**

- The **C pre-processor** has a **conditional clause** that tells the **C processor** whether it should *skip* the **processing** of a **portion of the C program file**

- The **C pre-processor** has the following **3 conditional** clauses:

- **#if**
 - **#ifdef**
 - **#ifndef**

The clause **#if** clause is *rarely used*....

I will skip it.

- **The #ifndef conditional**

- **Syntax of the #ifndef conditional:**

Syntax	Meaning
<pre>#ifndef MACRO-NAME text segment #endif</pre>	<ul style="list-style-type: none"> ■ If the MACRO-NAME is <i>defined</i>, then the text segment will be kept (not removed) by the C pre-processor ■ Otherwise, (the MACRO-NAME is <i>not defined</i>), then the text segment will be omitted (removed) by the C pre-processor
<pre>#ifndef MACRO-NAME text segment 1 #else text segment 2 #endif</pre>	<ul style="list-style-type: none"> ■ If the MACRO-NAME is <i>defined</i>, then the text segment 1 will be kept (not removed) <i>and</i> the text segment 2 will be omitted (removed) by the C pre-processor ■ Otherwise, (the MACRO-NAME is <i>not defined</i>), then the text segment 2 will be kept (not removed) <i>and</i> the text segment 1 will be omitted (removed) by the C pre-processor

```

#ifdef MACRO-NAME-1
    text segment 1
#elif MACRO-NAME-2
    text segment 2
[#else
    text segment 3]
#endif

```

- If the **MACRO-NAME-1** is *defined*, then the text segment 1 will be **kept (not removed)** and all other text segments will be **omitted (removed)** by the C pre-processor
- **Otherwise**, if the **MACRO-NAME-2** is *not defined*, then the text segment 2 will be **kept (not removed)** and all other text segments will be **omitted (removed)** by the C pre-processor
- **And so on.**

○ **Example:**

```

#define COMPILER_IT (This defines COMPILER_IT to be the empty string !)

#ifdef COMPILER_IT

    You can put any text here.
    The C-preprocessor will include it
    when the symbol COMPILER_IT
    is defined

#else

    You can put any text here.
    The C-preprocessor will include it
    when the symbol COMPILER_IT
    is NOT defined

#endif

```

Output of the C pre-processor:

```

You can put any text here.
The C-preprocessor will include it
when the symbol COMPILER_IT
is defined

```

○ **Example Program:** (Demo above code)

Example

- Prog file: [click here](#)

How to run the program:

- **Right click** on link and **save** in a scratch directory
- To compile: `gcc -E cprep1.c`
(The **-E** option tells the **C compiler** to **quit** after running the **C pre-processor**)

○ **Experiment:**

- **Delete** the line:

```
#define COMPILE_IT
```

- "Re-compile" the program file with:

```
gcc -E cprep1.c
```

- You will see the **text** in the **else part**

• **Defining symbol using the C-compiler option**

- The **C-compiler** has an **option** to **define** a symbol:

```
gcc -Dmacro[=defn] .....
```

○ **Illustrative example:**

- **Program:** `loveOrhate.c`

```
#ifdef LOVE
    Dear Abby,
    I love your column from yesterday....
#else
```

```

Dear Abby,

I hate your column from yesterday....

#endif

```

- Output of `gcc -E loveOrhate.c`:

```

Dear Abby,

I hate your column from yesterday....

```

because the **symbol** `LOVE` was *not* defined

- **However**, the output of `gcc -E -DLOVE loveOrhate.c`:

```

Dear Abby,

I love your column from yesterday....

```

because the **symbol** `LOVE` was *defined* by the compiler option `-DLOVE`

- **Example Program:** (Demo above code)

Example

- Prog file: [click here](#)

How to run the program:

- **Right click** on link and **save** in a scratch directory
 - To compile: **see above**

- **The `#ifndef` conditional**

- `#ifndef`:

- The `#ifndef` conditional is **similar** to the `#ifdef`, except the **meaning** is *reversed*:

```
#ifndef MACRO-NAME  
    text segment  
#endif
```

the `text segment` is **kept** when the macro name `MACR-NAME` is **not defined**

The `text segment` is **omitted otherwise** (when the macro name `MACR-NAME` is **defined**)

