

# C Preprocessor

- has more capabilities than we will cover
- **We will focus on this subset:**
  - including text file into your source code
    - `#include <filename>` - standard library
    - `#include "filename"` - own directory library
  - Macro Substitution (with or without parameters)
    - a.k.a. Symbolic Constants (a symbol for a constant value)
    - e.g., `#define EQUAL ==`
    - would allow you to use `EQUAL` in place of `==`

# File Inclusion

- Does text inclusion - includes entire contents of file in your source code at the point of #include statement BEFORE it goes to the compiler

to include commonly used code, declarations

- #include "filename"
- searches your directory for the file to include
- #include <stdio.h>
- searches established list of system directories for file to include
- .h is naming convention to indicate header file

## #define (macro substitution)

- macros (a.k.a. Symbolic Constants)

does straight text replacement

```
#define PI 3.14159
```

```
#define SECONDS_PER_DAY (60 * 60 * 24)
```

- #define macroname definition of macro
- - all on one line OR use line continuation \
- #define PRINT\_PROMPT printf("Prompt: ")
- parameterized macros
- parameters are placeholders for arguments require NO declaration
- #define MIN(x,y) (((x) < (y)) ? (x) : (y))  
#define MIN4(a,b,c,d) min(min(a,b), min(c,d))

## #define

```
#define MAX_LOOPS 15
```

```
for( i = 1; i < MAX_LOOPS; ++i)
```

is replaced BEFORE compilation by

```
for( i = 1; i < 15; ++i)
```

### #define (macros w/params)

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

```
c = MAX( f + x, f + v );
```

-- is replaced BEFORE compilation by --

```
c = MAX(( f + x ) > ( f + v ) ? ( f + x ) : ( f + v ));
```

--- what happens here? ---

```
charge = MAX( fee + fixed_costs, fee + var_costs );
```

which is easier to read, the code before or after text replacement?

### #define (macros w/params)

```
#define SQUARE( x ) x * x
```

(the symbols in the () will be replaced by parameter)

so ...

```
sq = SQUARE ( z + 1 );
```

-- becomes --

```
sq = z + 1 * z + 1
```

so **what is wrong?** Does this do what you want?

How can you make sure that it will?  
(see previous example)