#include

- #include <stdio.h>  start looking for this file using the path
- #include "filename"  start looking in current dir
- #include <filename>

- The #include … line is replaced by the contents of
- A preferred way to tie the declarations together for a large program.

#define

- #define name
- #define name text
- #define PI 3.1415926 ; #define e 2.17  ERROR
- #define TRUE 0
- #define AND &
- #define copyright All right reserved. No part \}
- of this publication may be reproduce, \}
- stored in a retrieval system
- #define record struct
- Macro substitution: literally replace occurrences of name in the program with text
- name has the same form as variables

#define - cont.

- Usage:
  - Constants
  - Repetitive small block of code
  - Debug code
- #undef name
  - It is not an error if the name is not defined originally
# define (cont.)

- #define name(parameters) text
- Examples:
- #define MAX(a, b) (a>b? a: b)
  MAX( i++, j++ ) ==> i++ > j++ ? i++ : j++
- #define square(x) x*x Wrong
  square(x+2) ==> x+2 * x+2
- #define square(x) (x)*(x)  ok

Another example

```c
#define PRINT_SIZE(type) \n  printf("%s\l", sizeof(type))

int main(void) {
  PRINT_SIZE(char);
  PRINT_SIZE(unsigned char);
  PRINT_SIZE(int);
  PRINT_SIZE(short int);
  PRINT_SIZE(long int);
  PRINT_SIZE(long long int);
  PRINT_SIZE(long);
  PRINT_SIZE(float);
  PRINT_SIZE(double);
  PRINT_SIZE(long double);
  return 0;
}
```

Conditional Code

```c
#if define name
<code>
#endif

#if define name
<code>
#else
<code>
#endif
```

- Making portable code
- Prevent multi-inclusion of header files
- Debugging
Conditional Code (cont.)

```c
#if constant_expression_1
<code 1>
#else
<code 2>
#endif
```

Predefined Names

- `_LINE_`: current source line number
- `__FILE__`: filename of the file currently being compiled
- Used for debugging

Passing macros in compilation

```c
gcc file8.c -Dkuku=17 -o file8

main()
{
    printf("%d", kuku);
}
```

Acknowledgement