

EECS Discussion Week2

TA: Weiwei CHEN
weiwei.chen@uci.edu
eecs10@eecs.uci.edu

Lab Tips

- How to change the password
 - % passwd
- Open two terminals for work
 - File editing
 - File compilation
- Pico tips:
 - Cut (ctrl+k)
 - Paste (uncut, ctrl+u)
 - Current Position (ctrl+c)
 - Search (where is, ctrl+w)
 - Search multiple times: keep pressing ctrl+w, then enter

October 3, 10

W. Chen

2

A simple C program

- Program Comments
 - Start with /* end with */
 - Be ignored by the compiler
 - Should be use to
 - Document the code line
 - Structure the program code
 - Enhance the readability
- Preprocessor directive #include
 - Insert a header file into the source code
- Standard header file
 - Comes from the C standard library
 - Contains the declarations of standard types and functions for data input and output (e.g. printf(), scanf())

```
/* HelloWorld.c: our first C program */
/* author: Rainer Doemer */
/* modifications: */
/* 09/28/04 RD initial version */
#include <stdio.h>
/* main function */
int main()
{
    printf("Hello World!\n");
    return 0;
}
/* EOF */
```

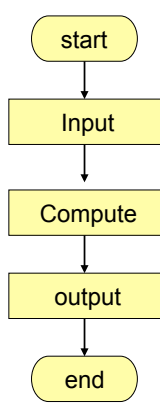
First C program

- int main (void)
 - Main function of the C program
 - Program entry (execution starts and ends)
 - Must return an integer (int) value to the operating system at the end of its execution
 - Return 0 indicates successful completion
 - Return value greater than 0 usually indicates an error condition
- Function body
 - Block of code (definitions and statements)
 - Starts with an opening brace ({)
 - Ends with a closing brace (})
- Printf() function
 - Formatted output (to stdout)
- Return statement
 - Ends a function and returns its arguments as result

```
...
/* main function */
int main(void)
{
    printf("Hello World!\n");
    return 0;
}
/* EOF */
```

Homework 2 (Part 1)

- Program structure
 - Input
 - read input data: scanf()
 - Compute
 - Compute output data from input data
 - Output
 - Write output data



```
graph TD; start([start]) --> Input[Input]; Input --> Compute[Compute]; Compute --> output[output]; output --> end([end]);
```

October 3, 10 W. Chen 5

Homework 2 (Part 1)

- Source code structure
 - Initialization section
 - Variable definition
 - Name, type, and initial value: `int number1 = 0;`
 - Input section
 - Read value from input device into variable
 - Standard input function: `scanf()`
 - Computation section
 - Perform the necessary computation on variables
 - Assignment statements: `number1 = 1 + 2;`
 - Output section
 - Write results from variable to output devices
 - Standard output function: `printf()`
 - Exit section
 - clean up and exit: `exit 0`

October 3, 10 W. Chen 6

Standard input/output functions

- Standard output function

```
#include <stdio.h>
```

```
int printf( const char *format, ... );
```

```
e.g. printf("The first number is %d. \n", number1);
```

- Standard input function

```
#include <stdio.h>
```

```
int scanf( const char *format, ... );
```

```
e.g. scanf("Number 1: %d", &number1);
```

October 3, 10

W. Chen

7

Conversion Specifiers for Basic Types

Type	printf()	scanf()
• long double	%Lf	%Lf
• double	%f	%lf
• float	%f	%f
• unsigned long long	%llu	%llu
• long long	%lld	%lld
• unsigned long	%lu	%lu
• long	%ld	%ld
• unsigned int	%u	%u
• int	%d	%d
• short	%hd	%hd
• Char	%c	%c

October 3, 10

W. Chen

8

Homework 2 (Part 2)

- Program structure
 - Input
 - read input data: scanf()
 - Compute
 - Compute output data from input data
 - Use arithmetic operations: +, -, *, /, %, <<, >>, ...
 - Be careful with the type of the variable.
 - Output
 - Write output data

October 3, 10 W. Chen 9

Homework 2 (Part 2)

- Carry over issues for summation
- Dividend = divisor * quotient + remainder
- Example: adding up two timestamps
 - Timestamp1: 43 seconds.
 - Timestamp2: 20 seconds.
 - Summation: (43+20) seconds. ?
 - Summation: (0 + 0 + 1) minutes, (43+20 -60) seconds.
 - 1 minutes, 3 seconds.

October 3, 10 W. Chen 10