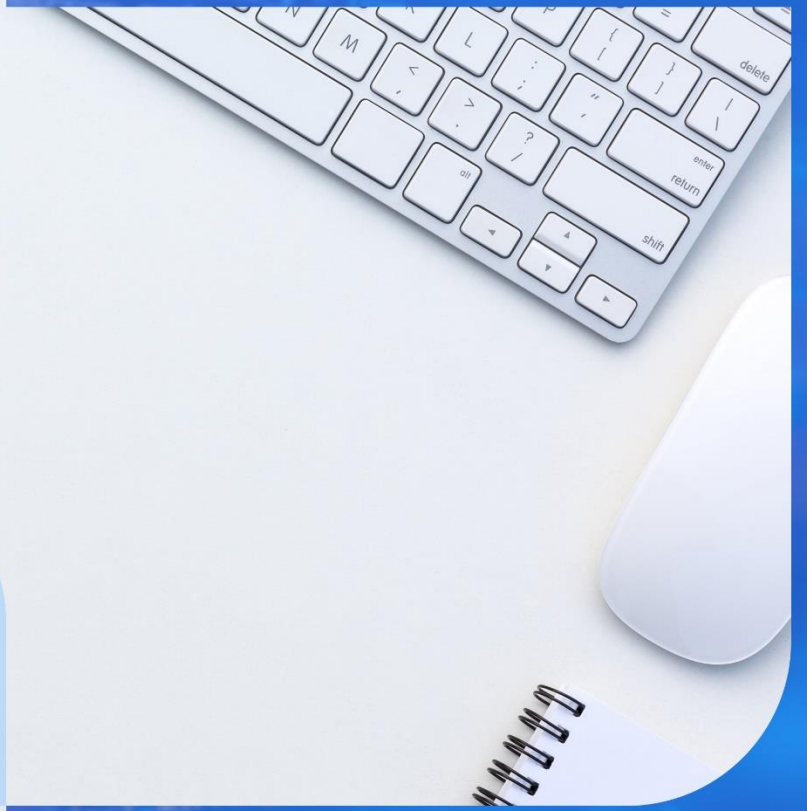
Basis and Practice in Programming

SUNG KYUN KWAN UNIVERSITY(SKKU)

Programming Basics

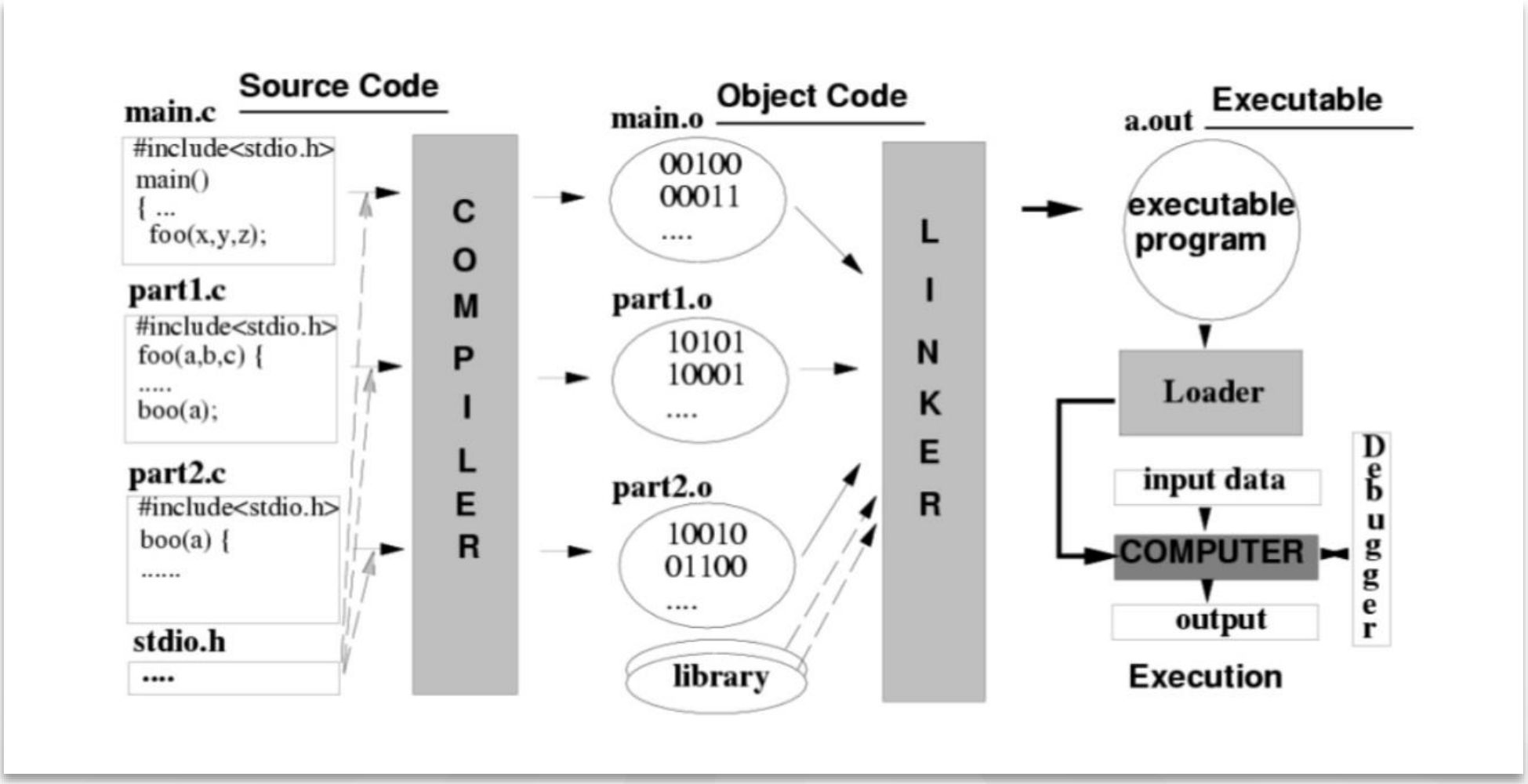


Instructor Younghoon Kim

A Simple C Program

Hello, World!

How a program works



A Simple C Program : Printing a Line of Text

SUNG KYUN KWAN UNIVERSITY(SKKU)

```
// Our first program in C
#include <stdio.h>

/* function main begins program execution
   there must be only one main function in your program! */
int main( void ) {
          printf("Hello, World!\n");
          return 0;
}
```

- This simple program will print out "Hello World!" on the screen.
- Contents inside of the figure of bottom-right corner is the output of the program.

A Simple C Program : Comments

SUNG KYUN KWAN UNIVERSITY(SKKU)

```
// Our first program in C
#include <stdio.h>

/* function main begins program execution
   there must be only one main function in your program! */
int main( void ) {
          printf("Hello, World!\n");
          return 0;
}
```

Hello, World!↵

- begin with //, indicating that this line is a *comment*.
- Comments document programs and improve program readability.
- Comments do not cause the computer to perform any action when the program is run.
- Comments are ignored by the C compiler and do not cause any machine-language object code to be generated.
- Multi-line comments in which everything from /* on the first line to */ at the end of the last line is a comment.

A Simple C Program : #include


```
// Our first program in C
#include <stdio.h>

/* function main begins program execution
   there must be only one main function in your program! */
int main( void ) {
          printf("Hello, World!\n");
          return 0;
}
```

Hello, World!↵

- `#include <stdio.h>` is a directive to the C preprocessor.
- Lines beginning with # are processed by the preprocessor before compilation.
- This line tells the preprocessor to include the contents of the standard input/output header (`<stdio.h>`) in the program.
- This header contains information used by the compiler when compiling calls to standard input/output library functions such as `printf`.

A Simple C Program : The main function

SUNG KYUN KWAN UNIVERSITY(SKKU)

```
// Our first program in C
#include <stdio.h>

/* function main begins program execution
   there must be only one main function in your program! */
int main( void ) {
          printf("Hello, World!\n");
          return 0;
}
```

Hello, World!↵

- The parentheses after main indicate that main is a program building block called a function.
- C programs contain one or more functions, one of which must be `main`.
- Every program in C begins executing at the function `main`.
- The keyword int to the left of main indicates that main "returns" an integer (whole number) value.
- Functions also can receive information when they’re called upon to execute.
- The void in parentheses here means that main does not receive any information.

A Simple C Program : The main function

SUNG KYUN KWAN UNIVERSITY(SKKU)

```
// Our first program in C
#include <stdio.h>

/* function main begins program execution
   there must be only one main function in your program! */
int main( void ) {
          printf("Hello, World!\n");
          return 0;
}
```

Hello, World!↵

- A left brace, {, begins the body of every function
- A corresponding right brace, }, ends each function
- This pair of braces and the portion of the program between the braces is called a block.

A Simple C Program : An Output Statement

SUNG KYUN KWAN UNIVERSITY(SKKU)

```
// Our first program in C
#include <stdio.h>

/* function main begins program execution
   there must be only one main function in your program! */
int main( void ) {
          printf("Hello, World!\n");
          return 0;
}
```

Hello, World!↵

- An output statement instructs the computer to perform an action, namely, to print on the screen the string of characters marked by the quotation marks.
- The entire line, including the `printf` function, its argument within the parentheses and the semicolon (`;`), is called a statement.
- Every statement must end with a semicolon (also known as the statement terminator).

A Simple C Program : Escape Sequences

SUNG KYUN KWAN UNIVERSITY(SKKU)

```
// Our first program in C
#include <stdio.h>

/* function main begins program execution
   there must be only one main function in your program! */
int main( void ) {
          printf("Hello, World!\n");
          return 0;
}
```

Hello, World!↵

- Notice that the characters \n were not printed on the screen.
- The backslash (\) is called an escape character -- `printf` is supposed to do something out of the ordinary.
- When encountering a backslash in a string, the compiler looks ahead at the next character and combines it with the backslash to form an escape sequence.
- The escape sequence \n means newline -- when a newline appears in the string output by a `printf`, the newline causes the cursor to position to the beginning of the next line on the screen.

A Simple C Program : Escape Sequences

SUNG KYUN KWAN UNIVERSITY(SKKU)

```
// Our first program in C
#include <stdio.h>

/* function main begins program execution
   there must be only one main function in your program! */
int main( void ) {
          printf("Hello, World!\n");
          return 0;
}
```

Hello, World!↵

- Because the backslash has special meaning in a string, i.e., the compiler recognizes it as an escape character, we use a double backslash (\\) to place a single backslash in a string.
- Printing a double quote also presents a problem because double quotes mark the boundaries of a string—such quotes are not printed.

A Simple C Program : Escape Sequences

```
// Our first program in C
#include <stdio.h>

/* func
   ther
int mai


}
```

orld!↵

Escape sequence	Description
\n	Newline. Position the cursor at the beginning of the next line.
\t	Horizontal tab. Move the cursor to the next tab stop.
\a	Alert. Produces a sound or visible alert without changing the current cursor position.
\\	Backslash. Insert a backslash character in a string.
\"	Double quote. Insert a double-quote character in a string.

- Notic
- The b ordinary.
- When encountering a backslash in a string, the compiler looks ahead at the next character and combines it with the backslash to form an escape sequence.
- The escape sequence \n means newline -- when a newline appears in the string output by a printf, the newline causes the cursor to position to the beginning of the next line on the screen.

A Simple C Program : The Linker and Executables

- Standard library functions like `printf` and `scanf` are not part of the C programming language.
- When the compiler compiles a `printf` statement, it merely provides space in the object program for a “call” to the library function.
- When the linker runs, it locates the library functions and inserts the proper calls to these library functions in the object program.
- Now the object program is complete and ready to be executed.
  - For this reason, the linked program is called an executable.
- If the function name is misspelled, it’s the linker that will spot the error, because it will not be able to match the name in the C program with the name of any known function in the libraries.

Variations on our first program

- Variations and some error cases will be shown during the demo.


```c
// Our first program in C
#include <stdio.h>

int main( void ) {
    printf("Hello,");
    printf(" World!\n");
    return 0;
}
```

```c
// Our first program in C
#include <stdio.h>

int main( void ) {
    printf("Hello,\n\nWorld\n!\n");
    return 0;
}
```

Follow-up question

- Make a program that prints out the following.

Hello, "Young"!↵