

# C++ BASICS & DATA TYPES

5<sup>TH</sup> WEEK LECTURE

엄현상(Eom, Hyeonsang)  
School of Computer Science and Engineering  
Seoul National University

# Outline

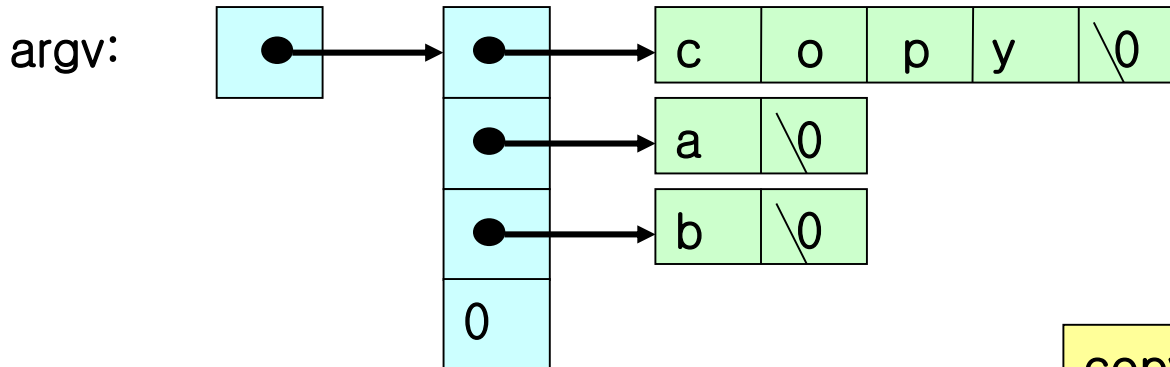
- C Programming Basics
  - Command-Line Argument Example
  - C input/output
- Q&A

# Command-Line-Argument Example

```
main(int argc, char *argv[]) {
```

```
...
```

```
martini:~>copy a b
```



```
for (int i = 0; i < argc; i++)  
    printf("%s %c \n", *(argv + i), *argv[i]);  
printf("%c \n",*++argv[0]);  
printf("%c \n",*++(argv[0]));  
printf("%c \n",(*++argv)[0]);  
printf("%c \n",*++argv[0]);
```

copy c  
a a  
b b  
o  
p  
a

# Input/Output: <stdio.h>

- To Permit Handling Buffer Allocation, and Performing I/O in Optimal-Sized Chunks
  - Stream (File Pointer): e.g., Standard Input
    - Buffering (Standard I/O Buffer; cf, Buffer Cache)
      - Full Buffering (\_IOFBF)
      - Line Buffering (\_IOLBF)
      - No Buffering (\_IONBF)

Standard output buffer flush;  
fflush(NULL) for all output streams

```
#include <stdio.h>  
...  
fflush(stdout); /* return EOF for a write error, and 0 otherwise */
```

Buffer cache flush;  
sync() for all modified block buffers

```
#include <unistd.h> /* optional */  
...  
fsync(1); /* return 0 if OK, and -1 otherwise */
```