



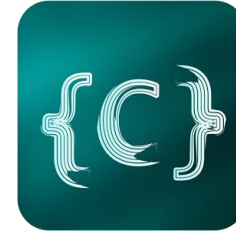
Hacettepe University

Computer Engineering Department

Programming in



python &



BBM103 Introduction to Programming Lab 1

Week 13

Fall 2017

C - Command Line Arguments

The command line arguments are handled using main() function arguments:

- **argc** refers to the number of arguments passed,
- **argv[]** is a pointer array which points to each argument passed to the program

Example :

```
#include <stdio.h>

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

    if( argc == 2 ) {
        printf("The argument supplied is %s\n", argv[1]);
    }
    else if( argc > 2 ) {
        printf("Too many arguments supplied.\n");
    }
    else {
        printf("One argument expected.\n");
    }
}
```

- When the above code is compiled and executed with single argument

```
./a.out testing
```

Output:

The argument supplied is testing

Example :

```
./a.out "testing1 testing2"
```

Program name ./a.out

The argument supplied is testing1 testing2

```
#include <stdio.h>

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

    printf("Program name %s\n", argv[0]);

    if( argc == 2 ) {
        printf("The argument supplied is %s\n", argv[1]);
    }
    else if( argc > 2 ) {
        printf("Too many arguments supplied.\n");
    }
    else {
        printf("One argument expected.\n");
    }
}
```

C - Functions

A function is a group of statements that together perform a task. Every C program has at least one function, which is **main()**, and all the most trivial programs can define additional functions.

Defining a Function

- The general form of a function definition in C programming language is as follows:

```
return_type function_name( parameter list ) {  
    body of the function  
}
```

Example:

```
/* function returning the max between two numbers */
int max(int num1, int num2) {

    /* local variable declaration */
    int result;

    if (num1 > num2)
        result = num1;
    else
        result = num2;

    return result;
}
```

Function Declarations

- A function **declaration** tells the compiler about a function name and how to call the function. The actual body of the function can be defined separately.

```
return_type function_name( parameter list );
```

```
int max(int num1, int num2);
```


Calling a Function – Example :

```
#include <stdio.h>

/* function declaration */
int max(int num1, int num2);

int main () {

    /* local variable definition */
    int a = 100;
    int b = 200;
    int ret;

    /* calling a function to get max value */
    ret = max(a, b);

    printf( "Max value is : %d\n", ret );

    return 0;
}
```

```
/* function returning the max between two numbers */
int max(int num1, int num2) {

    /* local variable declaration */
    int result;

    if (num1 > num2)
        result = num1;
    else
        result = num2;

    return result;
}
```

Output:

Max value is : 200