

C Programming Language

Hello World

Dr. Mostafa A. Elhosseini

Revise

- ✧ C history
- ✧ Features of C
- ✧ ANSI standard C
- ✧ C and C++
- ✧ C program development
- ✧ Standard input / standard output

Agenda

- ✂ Hello world layout
- ✂ Function
 - Parenthesis
 - Argument
 - void as an argument
 - Int as a return type
 - return
- ✂ Block
 - braces
- ✂ Header file
- ✂ String constant

Hello world example

- ✧ All C programs are divided into units called **functions**
- ✧ A function is like **subroutine** in BASIC or a function in PASCAL
- ✧ Note that **main** is a function
- ✧ Every C program consists of one or more functions
 - **main** is the one to which **control is passed from operating system**
- ✧ **int** preceding main specifies that function main will return a value
- ✧ **void** in parenthesis specifies that the function **takes no argument**

```
#include <stdio.h>

int main (void)
{
    printf ("I am FasheL");
    return 0;
}
```

function name

opening brace

one statement

Semicolon terminate each program statement

closing brace

Hello world example

- ✧ Space, tab, newline; **whitespace** characters are **invisible to the compiler**
- ✧ **Indentation** of blocks of code enclosed in braces makes program more readable
- ✧ Printf() is a **function** name
 - Since printf() is a function, it is **followed by parenthesis**
- ✧ The phrase **"I am fashel"** is the function **argument**
 - Information passed from main() to printf()
- ✧ **"I am fashel"** is a string constant – surrounded by quotes
- ✧ The code that describe how printf work does not exist in the source file
 - It is in the **header** file stdio.h
- ✧ C is case sensitive

```
#include <stdio.h>

int main (void)
{
    printf ("I am fashel");
    return 0;
}
```

function name

opening brace

one statement

closing brace

Semicolon terminate each program statement

Hello world example

- ✗ **return** causes the value, here **0**, to be passed back to the operating system
 - 0 means success
 - A value of 1, 2, 3 etc. indicates failure
- ✗ **Semicolon** indicate statement terminators
 - Tell the compiler where one statement ends and the next one begins
- ✗ Having stated that main is to return an integer to the operating system, forgetting to do so either by saying **return only**, or by **omitting the return** would cause a random integer to be returned to the operating system
 - Random value could be zero "success"
 - Non-zero value randomly indicate failure

```
#include <stdio.h>

int main (void)
{
    printf ("I am FasheL");
    return 0;
}
```

function name

opening brace

one statement

closing brace

Semicolon terminate each program statement