Multi-Paradigm Programming - The C Programming Language

Dominic Carr

Atlantic Technological University

dominic.carr@atu.ie

What We Will Cover

1. Goals of this Session
2. The C Programming Language
3. Comparison with Python
4. Installing C on Windows
5. C Practice Questions

Goals of this Session

Goals

- To understand....
  - The basics of the C programming language
  - How to write a procedural program in C

The C Programming Language

The C Programming Language I

- General-purpose & procedural computer programming language
- Supports structured programming, lexical variable scope, and recursion
- Static type system prevents unintended operations.
- By design, C provides constructs that **map efficiently** to typical **machine instructions**
  - Has found lasting use in applications previously coded in **assembly language**.
  - Including operating systems and application software for diverse platforms from supercomputers to embedded systems.
  - It has been around since 1972
  - Was developed at Bell Labs

The C Programming Language II

Listing 1: Assembly Code for Hello World

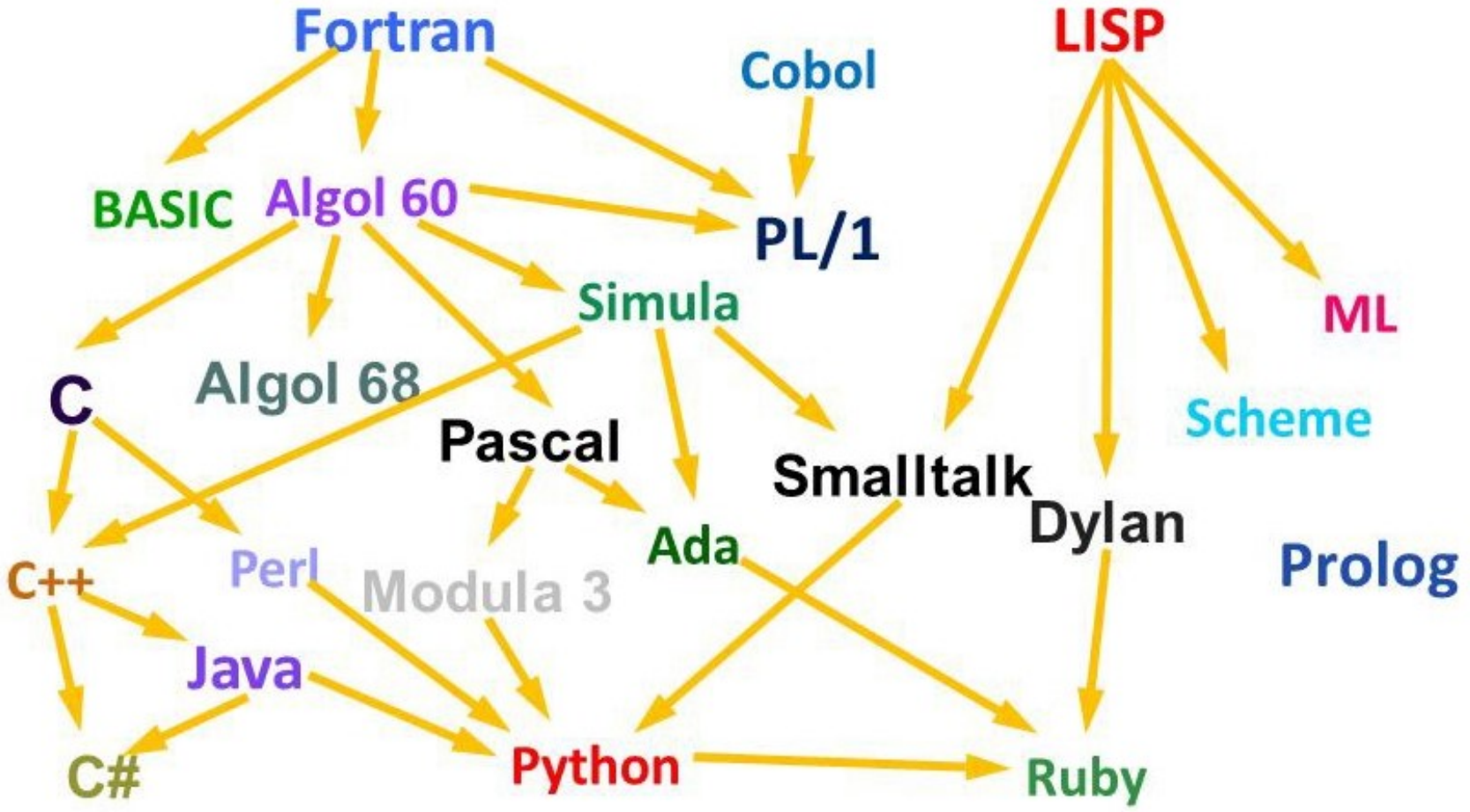
```
    global _main
    extern _printf

    section .text
_main:
    push   message
    call   _printf
    add    esp, 4
    ret
message:
    db  'Hello, World!', 10, 0
```

C was designed to replace this!

The C Programming Language III

- Designed to be compiled using a relatively straightforward compiler
- to provide *low-level access to memory* and language constructs that map efficiently to machine instructions.
- Designed to work cross-platform.
  - A standards-compliant C program written with portability in mind can be compiled for a wide variety of computer platforms with **minor changes** to its source code.
  - Java was designed with even better cross platform support under the tagline "Write once, run anywhere".
- The language is available on various platforms, from embedded microcontrollers to supercomputers.

The C Programming Language IV



The C Programming Language V

Listing 2: Hello World in Standards Compliant C

```
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
}
```

- "#include" is a pre-processing directive, it is saying to pull the contents of the specified file and replace this line with that.
  - "stdio.h" is part of Standard C.
- "main()" is a function, but it is a very special function
  - It acts as the entry point of the program it is from here that execution begins. Main returns an int to the calling environment.

The C Programming Language VI

- The next line calls (diverts execution to) a function named printf()
  - This is a function found in the C system library which sends output to the "standard out" of the calling environment (usually the terminal/command prompt)
    - That said std out can be redirected to funnel information between scripts or into files.
  - printf() will output the character array to the standard output.
  - We do not have to explicitly return a value for main it implicitly returns "0" which means the program executed successfully.

The C Programming Language VII

Q & A

Write a C program to print your name, date of birth, and mobile number.

```
#include <stdio.h>

int main(void)
{
        printf("Name: Dominic Carr\n");
        printf("DOB: June 12th, 1920\n");
        printf("086-1910000\n");
}
```

The C Programming Language VIII

Q & A

Write a C program to print a block F using hash (#), where the F has a height of six characters and width of five and four characters

```
#include <stdio.h>
int main()
{
        printf("######\n");
        printf("#\n");
        printf("#\n");
        printf("#####\n");
        printf("#\n");
        printf("#\n");
        printf("#\n");
}
```

The C Programming Language IX

Listing 3: Another answer this time with a function

```
#include <stdio.h>

void print(int times, char a)
{
        for(int i = 0; i< times; i++)
        {
                printf("%c", a);
        }
        printf("\n");
}

int main()
{
        print(6,'#');
        print(1,'#');
        print(1,'#');
        print(5,'#');
```

The C Programming Language X

```
        print(1,'#');
        print(1,'#');
        print(1,'#');
}
```

The C Programming Language XI

- Write a C program to compute the sum of the two given integer values. If the two values are the same, then return triple their sum.

```
#include <stdio.h>

int sum(int a, int b)
{
        if (a==b)
        {
                return (a+b)*3;
        } else {
                return (a+b);
        }
}

int main()
{
        int res = sum(1,2);
```

The C Programming Language XII

```
        printf("Result is %d\n", res);
        res = sum(3,3);
        printf("Result is %d\n", res);
}
```

Comparison with Python

Comparison with Python I

- C is compiled, Python is interpreted
- C allows low level memory access, Python does not
  - Different levels of abstraction
- Python support OOP, C does not
- Python has a much larger set of built-in functionality than C
- C code execution is much faster than Python
  - Key for certain systems
  - Compilation is major factor

Comparison with Python II

- Variable types must be declared in C, not so in Python
  - Static vs dynamic typing
- C has a more verbose syntax than Python
  - Python would be considered easier to learn
- In C memory management is manual, Python has automated management
- There are many syntactical differences, but some commonalities

Installing C on Windows

Installing C on Windows

Several different ways, one is Cygwin:

- Install Cygwin, which gives us a **Unix-like environment** running on Windows.
- Install a set of Cygwin packages required for building **GCC**.
- From within Cygwin, download the GCC source code, build and install it.
- Then you should be able to compile and run C programs.

Note: Your VM for this module has the C Compiler pre-installed

- For quick tests you can use the online compiler @ `https://www.onlinegdb.com/online_c_compiler`

C Practice Questions

C Practice Questions I

We will take a look at some of the "elementary" questions @ `https://adriann.github.io/programming_problems.html`

The End