Multi-Paradigm Programming

Dominic Carr

Galway-Mayo Institute of Technology

dominic.carr@gmit.ie

What We Will Cover

1. Goals of this Session

2. The C Programming Langauge
   - Structuring Data in C
   - Comparison with Python
   - Installing C on Windows
   - C Practice Questions

3. Sources

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 Langauge

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
```

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 and operating systems with **few 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

Listing 2: Hello World in Standards Compliant C

```c
#include <stdio.h>

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

The C Programming Language V

- “#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 next line calls (diverts execution to) a function named printf
  - This is a function found in the system library of C which sends output to the “standard out” of the calling environment, typically this means it prints out to the terminal or 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 imiplictly returns “0” which means the program executed successfully.

The C Programming Language VI

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 VII

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 VIII

Listing 3: Another Answer this time with a function

```c
#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 IX

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

The C Programming Language X

- 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 XI

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

Structs I

Listing 4: C Example of representing a Person

```
#include <stdio.h>
struct person
{
   int age;
   float weight;
};
int main()
{
    struct person *personPtr, person1;
    personPtr = &person1;
    printf("Enter age: ");
    scanf("%d", &personPtr->age);
    printf("Enter weight: ");
    scanf("%f", &personPtr->weight);
    printf("Age: %d\n", personPtr->age);
    return 0;
```

Structs II

```
}
```

Structs III

- A struct is a way of grouping individual variables together
- It can be used to create a representation of something like a person
- As defined above a person has an age and a weight
- We will contrast this with OOP approaches which we will learn about soon

C vs Python

- C is compiled, Python is interpreted
- C allows low level memory access, Python does not
- 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
  - Compilation is key
- Variable types must be declared in C, not so in Python
- C has a more verbose syntax than Python
  - Python would be considered easier to learn
- In C memory manaegement is manual, Python has automated management
- There are many syntactical differences, but some commonalities

Installing C on Windows

- 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.

Follow these steps which are detailed extensively @ `https://preshing.com/20141108/how-to-install-the-latest-gcc-on-windows/`

Online C Compiler

`https://www.onlinegdb.com/online_c_compiler` and Compile our C code online!

C Practice Questions I

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

C Practice Questions II

Questions:

- Write a C program to get the absolute difference between n and 51. If n is greater than 51 return triple the absolute difference.
- Write a C program to check two given integers, and return true if one of them is 30 or if their sum is 30.
- Write a C program to compute the sum of the two given integers. If the sum is in the range 10..20 inclusive return 30
- Write a C program that accept two integers and return true if either one is 5 or their sum or difference is 5.
- Write a C program to check if y is greater than x, and z is greater than y from three given integers x,y,z
- Write a C program to check if two or more non-negative given integers have the same rightmost digit.

Sources

Sources

- https://www.computerhope.com/jargon/i/imp-programming.htm
- https://en.wikipedia.org/wiki/Abstraction_(computer_science)
- https://www.w3resource.com/c-programming-exercises/basic-algo/index.php

The End