

# WELCOME TO CSCUV4 PRACTICALS - WEEK 2

*Cristian Lepore*

# PLAN FOR TODAY

- 12 – 12:20 >> presentation
- 12:20 – 12:30 >> Q&A
- 12:30 – 1pm >> exercises & checkpoints

# FIRST PART

- Basic concepts
- First program – Hello world
- Printf
- Scanf

# MAIN CONCEPTS

- C is a **procedural language** is a computer programming language that follows, in order, a set of commands. Examples of computer procedural languages are Basic, C, Fortran, and Pascal.
- Same types of control flow structures as Java:
  - for, while, do-while, if, switch.
- Hello World comes from the C programming language.


# MAIN FUNCTION

```
1  
2  
3 int main() {  
4  
5     return 0;  
6 }  
7
```

---

# RETURN VALUE

```
1  
2  
3 int main() {  
4  
5     return 0;  
6 }  
7
```



Return 0 : termination  
without error

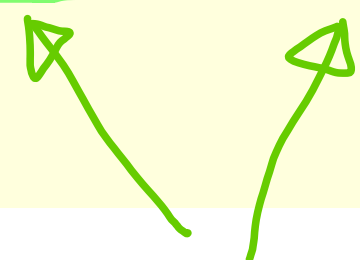


Value greater than 0 :  
termination with error

# COMPILE OUR FIRST PROGRAM

- Several compiler for C under Unix and Windows.
- GCC compiler is the open source project

```
clepore@Device:~/Desktop$ gcc main.c
clepore@Device:~/Desktop$ ./a.out
Hello world!
\clepore@Device:~/Desktop$ gcc -Wall main.c -o myExe
clepore@Device:~/Desktop$ ./myExe
Hello world!
\clepore@Device:~/Desktop$ █
```



# C COMPILER IS QUITE PERMISSIVE COMPARED WITH JAVA

```
1
2
3 int main() {
4     int x;
5     x = 34.2;
6         |
7     return 0;
8 }
9 |
```

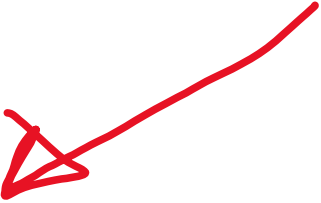


# VARIABLE DECLARATION

```
1  #include <stdio.h>
2
3  int main() {
4      int i=0;
5
6      for(; i<10; i++){
7          /* Do something */
8      }
9
10     return 0;
11 }
12
```

```
1  #include <stdio.h>
2
3  int main() {
4
5      for(int i=0; i<10; i++){
6          /* Do something */
7      }
8
9      return 0;
10 }
11
```

## <STDIO.H>



```
1  #include <stdio.h>
2
3  int main() {
4
5      printf("Hello world!");
6      printf("\n \t \\");
7
8      return 0;
9  }
10
```

# PRINTF

```
1  #include <stdio.h>
2
3  int main() {
4      int i = 6;
5      int j = 2;
6
7      printf(" %d %d ", i, j);
8
9      return 0;
10 }
11 |
```

# MACRO

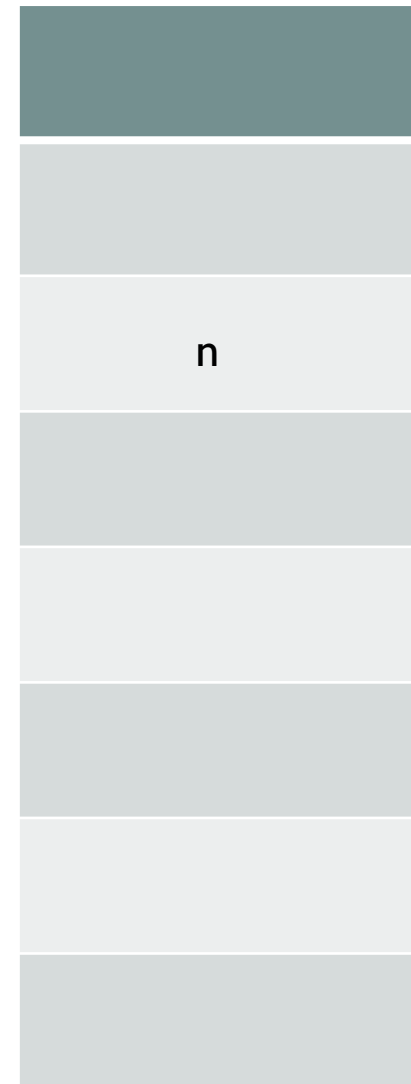
```
1  #include <stdio.h>
2  #define TRUE 1
3  #define FALSE 0
4
5  int main() {
6      |
7      return 0;
8  }
9
```

# SCANF (1)

```
1  #include <stdio.h>
2
3  int main() {
4      int n;
5      scanf("%d", &n);
6
7      char str[] = "Hello World";
8
9      return 0;
10 }
11
```

Address: 123

MEMORY



...

## SCANF (2)

```
1  #include <stdio.h>
2
3  int main() {
4      int n;
5      scanf("%d", &n);
6
7      char str[] = "Hello World";
8
9      return 0;
10 }
11
```

Address: 122

Address: 123

...

Address: 126

Address: 127

...

### MEMORY

Address: 126

H

e

l

...



## MAIN FUNCTION (1)

1 \*


2

```
3 int main() {  
4  
5     return 0;  
6 }  
7
```

1

2

```
3 int main(int argc, char *argv[]) {  
4  
5     return 0;  
6 }  
7
```



## MAIN FUNCTION (2)

```
~/Desktop$ ./a.out 12 23 "Hello World"█
```

0 1 2 3

Argc = 4 --> the number of arguments

Argv[2]= 23 --> the second argument



ANY QUESTIONS?