Embedded Software
CS 145/145L

Caio Batista de Melo
Defining Blocks

- whitespace doesn’t matter
- blocks are defined by {}

```python
if True:
    pass
    # block of code here
```

```python
if (1) {
    // block of code here...
    // and same block here...
    // and here...
}
```

But you should still try to make your code look nice!
Variables

- static typing;
- need to declare everything beforehand!

\[
\begin{align*}
x &= 2 \\
y &= 2 \times x \\
x &= 2.0 \\
z &= 2 \times x
\end{align*}
\]
Functions

- functions also need types (i.e., return type);
- parameters also need types;
- if you want to modify the arguments externally (side-effects), you need pointers!

```python
def square_num(x):
    return x * x

int square_num(int x) {
    return x * x;
}
```
Pointers

```c
void square_nums_inplace(int* x, int y) {
    *x = *x * *x;
    y = y * y;
}

int main() {
    int x = 2, y = 2;
    square_nums_inplace(&x, y);
    // x == 4 after this, but y is still 2!
    return 0;
}
```
Arrays

- homogeneous typing
- need to declare size beforehand!
- strings are just char arrays, so same restrictions!
- for strings, need to make the last element == 0

```python
my_lst = []
my_lst.append(10)
my_lst.append('hello')
my_lst.append(2.5)
```
Loops

- for loops are similar to for in range(), there is no *for-each*
- while loops are pretty similar

\[
x = 1 \\
\text{for } i \text{ in range}(1, 10): \\
\quad x *= i
\]

\[
\text{int } x = 1; \\
\text{int } i; \\
\text{for } (i=1; i < 10; i++) \\
\quad \{ \\
\quad \quad x *= i; \\
\quad \}
\]
your code starts in the \textit{int main} function (like the main-block);

it should return 0 to tell the computer that the program exited without errors

\begin{itemize}
  \item although in our projects we should never reach the return!
\end{itemize}

\begin{Verbatim}
def square_num(x):
    return x * x

if \texttt{\_\_name\_\_} == \texttt{\_\_main\_\_}:
    square_num(2)
\end{Verbatim}

\begin{Verbatim}
int square_num(int x) {
    return x * x;
}

int main() {
    square_num(2);
    return 0;
}
\end{Verbatim}
Source Files

Python source file (square.py)

```python
def square_num(x):
    return x * x
```

C header file (square.h)

```c
#include "square.h"

int square_num(int x) {
    return x * x;
}
```

C source file (square.c)
Includes

- include the .h files;
- use <> for system-library headers, "" for your own custom headers.

```c
#include "square.h"

int square_num(int x) {
    return x * x;
}

#include <avr/interrupt.h>
#include <avr/pgmspace.h>
#include <avr/io.h>
```

square.h

```c
#include <avr/interrupt.h>
#include <avr/pgmspace.h>
#include <avr/io.h>
```

avr.h
Further Reading

Python to C
- [https://www.cs.toronto.edu/~patitsas/cs190/c_for_python.html](https://www.cs.toronto.edu/~patitsas/cs190/c_for_python.html)
- [https://realpython.com/c-for-python-programmers/](https://realpython.com/c-for-python-programmers/)

Generic C Material
- [https://www.learn-c.org](https://www.learn-c.org)
- [https://www.programiz.com/c-programming](https://www.programiz.com/c-programming)
- [http://www.faqs.org/docs/learnC/](http://www.faqs.org/docs/learnC/)
See you next time :)  

Q & A