ENCE260

C language reference

Okay, so the test is coming up (Monday 2nd September), and it's open book, so I made this as part of my study. Hopefully it can be of some use to someone else.

Before getting into it, here's some more information about the test...

The test will be in the labs (mainly lab 2, overflow in others).

You need to book a time. See learn for info' on that. There are two time slots available (5pm, 7pm).

It is done in a special linux environment. You will have access to Geany and the quiz server. No Google, no Facebook, etc. The test is open, but not that open.

The test will be quiz-format, and so expect something similar to a lab.

One and a half hours is allowed. The time remaining will display on the quiz page.

What is C?

You should know this already. It's a programming language. This is simply a reference, and so I'm going to assume that trivial things like this are known, so don't expect this to be a complete introduction. :D

CPP

CPP is the C pre-processer and does all of the things (preprocessor directives) that begin with a hash

#include <stdio.h>
- This includes the contents of stdio.h. Just copy and paste, really. The < >'s indicate that it's a built-in function. Use double quotes for any locally-included headers.

#define Sequence1 Sequence2
- A find-and-replace at compile time. Use these to make your code more readable and easier to maintain.

#ifdef / #ifndef
- Do something if something is/is not defined. Great for avoiding multiple definitions (such as importing a header multiple times).
If blocks must end with #endif

Those are the main 3 (and only 3 that we covered), but others exist, which undefine things things that aide control flow like else.

Naming

Be descriptive but concise.

Only use single letters if they're obvious and local.

s for a string in a small function is fine. But s for size in a 300-line program is pushing it. Those 3 bytes you're saving on your computer's disk aren't worth the confusion.

Typing

Unlike python, c is explicitly typed. You need to specify the type of something when you make it.

```
int number;
char c;
float* f;
```

Assignment

Evaluates RHS. Puts it in LHS. That's the nuts of it. There's more to it then that, but for simplicity, assume that's how it works.

```
int number;
number = 10;
char c1 = 5;
char c2 = 't';
```

Assignment can also be used in conjunction.

```
c1 = c2 = 'q';
```

Operations

The same as what you know and love from whatever languages you've learned previously.

```
+    - addition
-    - subtraction
*    - multiplication
/    - division
++   - increment
--   - decrement
```

There are also bitwise operators but they haven't been covered and so will likely not be needed for this test.

For interest sake, though...

```
<<   - left bitshift
>>   - right bitshift
```

Conditional operators are also a thing which are used in conditions, such as control flow (if-, while-, ...)

```
&&   - Logical AND
||   - Logical OR
!    - Logical NOT
&    - Bitwise AND
|    - Bitwise OR
^    - Bitwise XOR
```

To see the difference between logical and bitwise, consider 5 & 2

```
1001 - 5
1010 - 6
&----
1000 - 4
5 && 6 = 1
```

Because both 5 and 6 are not zero, the logical and gives true (represented by 1).