

C243 Data Structures

Programming Style Requirements

Indentation and Spacing

- The indentation will be set at 4 spaces precisely.
- The indentation must be done using spaces and not tabs.
- Each instruction must be placed on its own line, including curly braces.
- The braces marking a block must be aligned with each other and everything else must be inside.
- An else followed by another if can be placed on a single line.
- An empty line must separate each function from the next one.
- No line must be longer than 100 characters, including comments.
- If you must break a line of code, use "smart alignment": either align the first item on the second line with a similar item on the line above, or just add 4 extra spaces so we can see that it's a new line.

Naming Conventions

- Names must be well enough defined to make it clear what their purpose is.
- We'll use CamelCase for names composed of more than one word.
- Variable, parameter, and function names must start with a lowercase letter, except for constructors.
- Class and type names must start with an uppercase letter.
- Constants must be in all uppercase.
- If you use an abbreviation, such as "num" for number, use the same one for all the names containing this word.

Program Structure

- Reasonable expectations concerning the program structure and clarity.
- Functions should be commented:
 - one comment at the top stating what the function does, what the parameters should be, and what kind of value it returns;
 - a comment with the name of the function at the end;
 - other comments that can explain the functionality inside the body of the function.
- The conventional length of functions is 20 lines of code, although this will not be strictly enforced, especially when they contain a long switch statement.

Compilation

- Multiple source files will be very common.
- The submitted code should compile on Linux with g++, which is the current setting in our labs.
- Using Visual Studio is not recommended. However, if you choose to use it, you must verify that the program compiles with g++ before submitting it, and you must make sure that the editor is set up correctly for the spacing and the indentation.

Setting the Indentation

Here are some instructions for setting the indentation to 4 spaces and not to use tabs by editor.

Emacs

Edit a file called .emacs (the dot is part of the name) directly in your home directory (create it if it doesn't exist) and add the following lines at the end:

```
; use spaces instead of tabs
(setq-default indent-tabs-mode nil)
; use 4 spaces in the indentation
(setq default-tab-width 4)
(setq tab-width 4)
(setq c-default-style "linux")
(setq c-basic-offset 4)
(c-set-offset 'comment-intro 0)
```

Visual Studio 2010 and up

From the Tools menu, click on Options, then the Text Editor tab. Click either the All Languages section, or C/C++.

Check the option Insert Spaces.

Set the Indent Size to 4 and also Tab Size to 4 (to make sure).

This may work similarly with other versions of VS.