First steps with C++

UNIX

creating, moving, renaming, copying, moving around in a file system, and EDITING (using your preferred editor!) (see also UNIX cheatsheet).

Reserved words

`http://cs.smu.ca/~porter/csc/ref/cpp_keywords.html`

Hello world!

A common first program is to print out a string ”Hello World”, one would think that this is easy in any language but it is actually astonishingly complicated in C++.

In Pseudocode I could write

```
print ”hello␣world”
```

Here a most minimalist C++ example (I suggest that you do not code like this!)

```
#include <iostream>
using namespace std;
int main() {
    cout << ”Hello␣World” << ’\n’;
    return 0;
}
```

the following example is certainly overkill, you will need to find a middle ground in documenting parts of the program

```
/*
 * First encounter with C++
 * program simply prints out a text
 * Version 1: barebones as possible
 *
 * (c) Peter Beerli, Tallahassee FL, 2016
 */

#include <iostream>

// name spaces are lazy prgrammer tricks
// the make programming easier, but in large projects
// they are more of a problem than help
// <iostream> is a standard package and therefore part of std
// namespace allows us to call it without specifying the std part
// we could rewrite without namespace ==> see helloworld2.cpp

using namespace std;

/*
```

```
 * Simply prints out the title 'Hello␣World'
 */
int main(int argc, char** argv) {
  //prints a newline
  cout << '\n';
  // prints the text
  cout << "Hello␣World" << '\n';
  // underlines the text
  cout << "===========" << endl;
  // the main function MUST return a integer value
  // commonly we return 0 on sucess; 1 on failure
  // but most commonly we do not care about the value that it returns
  return 0;
}
```

Remember that you should comment on shortcuts, unclear code (well! we should not have that!), or complex logic. Here is an example I find sufficient and that is not using a `using namespace`

```
/*
 * First encounter with C++
 * program simply prints out a text
 *
 * (c) Peter Beerli, Tallahassee FL, 2016
 * contact: beerli@fsu.edu
 */

#include <iostream>

// Simply prints out the title 'Hello␣World'
// this programs does not need any arguments
//
int main(int argc, char** argv) {

  std::cout << '\n';
  std::cout << "Hello␣World" << '\n';
  std::cout << "===========" << std::endl;

  return 0;
}
```

How to make it work?

Once we have written the code we need to translate it into code the hardware can understand.

```
g++ -o hello helloworld.cpp
```

g++ is the compiler (the program that translates your C++ code into 'byte' code, that can be loaded into the cpu and that then delivers an answer (if your code is correct and the compiler did its job!). The `-o` flag uses an argument that specifies the compiled output (the program). If you do not use ut the program is named by convention `a.out`

```
g++ helloworld.cpp
```

More things

```
/*
 * First encounter with C++
 * program simply prints out a text
 *
 * (c) Peter Beerli, Tallahassee FL, 2016
 * contact: beerli@fsu.edu
 */

#include <iostream>
#include <string>

// Simply prints out a message that is hardcoded into a string
// this programs does not need any arguments
//
int main(int argc, char** argv) {

  std::string message = ”The␣quick␣fox␣jumps␣over␣the␣lazy␣dog”;
  std::cout << ’\n’;
  std::cout << message << std::endl;
  return 0;
}
```