Week 12 — November 29, 2018

Homework

- Read pp. 1-9, pp. 20-22, pp. 315-333 in "Beginning C++"

Exercises

1. Write a "Hello World!" program in C++, i.e., a program that writes a string to the standard output. Compile and run your program.

   Remarks: C++ source files typically have the extension `.cpp` and `.h` for source code and header files, respectively. (Other conventions exist, e.g., `.cc`, `.cxx`, or `.c++` for source files and `.hh`, `.hxx`, or `.h++` for header files.) To explicitly use the GNU C++ compiler within the GCC, use `g++` instead of `gcc`. Similarly, if you use Clang, you may use `clang++` to explicitly invoke the C++ compiler. (On some systems you may also be able to use the generic command `c++`.) You can use the following makefile template:

   ```
   CXX=g++
   CPPFLAGS=
   CXXFLAGS=-Wall -std=c++11
   LDFLAGS=
   LDLIBS=
   LINK.o=$(CXX) $(LDFLAGS)

   ### Insert targets and prerequisites below
   # target: prerequisites

   .PHONY: clean
   clean:
       -$(RM) *.o
   ```

2. Write a program that prompts the user to input his or her name and age, and then writes a suitable welcome message to the user. Use a `string` to store the name and use an integer to store the age.

3. Do exercise 11-1 in "Beginning C++". You may use the following template:

   ```
   #include <iostream>

   class Integer {
   private:
      int i;               // value
   public:
      Integer();           // "constructor"
      void set(int newi);  // "setter"
   ```

```
    int get();           // "getter"
    void print();        // print value
};

Integer::Integer() {
  // insert code here
}

void Integer::set(int newi) {
  // insert code here
}

int Integer::get() {
  // insert code here
}

void Integer::print() {
  // insert code here
}

int main(int argc, const char *argv[]) {
  // insert code here
  return 0;
}
```

4. Do exercise 11-2 in "Beginning C++".
5. Do exercise 11-3 in "Beginning C++".

Optional exercises

- Watch Bjarne Stroustrup, the creator of C++, explain why he created C++.
- Read about C# (pronounced *C sharp*) and Objective-C which, like C++, are object-oriented programming languages that are based on C.

  Fun fact: the lead architect of C# is Anders Hejlsberg, a former DTU student.
- Watch Bjarne Stroustrup explain why he believes that the C language is obsolete.

  As always, there are two sides to every story. Linus Torvalds, the creator of the Linux kernel, is quoted as saying that *"C++ is a horrible language. It's made more horrible by the fact that a lot of substandard programmers use it, to the point where it's much much easier to generate total and utter crap with it. Quite frankly, even if the choice of C were to do nothing but keep the C++ programmers out, that in itself would be a huge reason to use C."*

  Now look at the Programming Community index that tracks the popularity of programming languages. Click on the different C-based languages (C, C++, C#, Objective C). Is the C language or its object-oriented extensions going to disappear any time soon?