1 Welcome

Key terms: C++
Reading: Course policies

CS110A is an introduction to programming using the famous C++ language. Programs exist to compute and store information at greater scale than is otherwise possible. Computers hosting certain application programs are used in nearly every field of endeavor. Your goal is to develop fluency in C++ and become aware of related computer science topics. A dedicated novice completing this course is fluent enough to reach a programmer’s first major milestone, writing a useful program.

While many elements of C++ are similar to those of other programming languages, it bears noticing that the language features build upon each other and are only useful in combination. Any topic you do not complete will put a big brake on your progress more generally.

The course covers the basics of programming logic. By its conclusion you will understand the kind of work a computer programmer does. This course on its own is not enough to become a professional programmer, but it can set you on your way to being one.

Student programming tutors staff the ACRC lab, where computers are available.

Below is a fragment of a C++ program, which you are not expected to understand now; you will, though, understand it at the course’s conclusion. Spend a few minutes looking at it and try to figure out what the various parts do. If you already understand this program, you should enroll in CS110B instead.

```cpp
bool palindromeCheck ( string word ){
    // compare pairs of letters and bail out if they fail to match
    for ( int i = 0 ; i < word.size() / 2 ; i ++ )
        if ( word [i] != word [word.size() - i - 1 ] )
            return (false);
    // all the pairs matched, so it was a palindrome
    return (true);
}
```

Make sure you have read and understood the course policies.

Ensure that you have easy access to the course textbook and to a machine of some kind, preferably with a keyboard. The course textbook is Allan Downey’s “How to Think like a Computer Scientist.”

In this course you will be challenged to understand cryptic techniques throughout. You will be using reference materials which contain organized explanations and a forum in which you and your peers can compare experiences and notes. Personal tutoring by more advanced students is frequently available in the ACRC lab in Batmale Hall. You are also welcome to come to your instructor’s office hours for advice or guidance. A galaxy of further reading is out there for all the topics we cover.

Next, we will discuss hills, the server on which we’ll write and run our programs.

---

1https://fog.ccsf.edu/~abrick/
2http://www.ccsf.edu/acrc
3http://fog.ccsf.edu/~abrick/Downey - How to Think Like a Computer Scientist (C++).pdf