

## **SAAST 2012 | Computer Science**

(Preliminary syllabus | subject to change)

This course is a fast-paced introduction to computer science by way of learning the fundamentals of computer programming in Python. The heart of computer science lies in algorithmic thinking or how to "think like a computer". While algorithmic thinking and computer programming are not the same thing, they are closely related as the famous computer scientist Donald Knuth once wrote:

It has often been said that a person does not really understand something until after teaching it to someone else. Actually a person does not really understand something until after teaching it to a computer, i.e., expressing it as an algorithm.

If you are interested in pursuing a degree in computer science, this course will develop your skills in the three areas of algorithmic thinking | precision, decomposition, and abstraction | using computer programs as a way to make all of these things concrete. Even if you aren't interested in computer science as a future major, you will find that the algorithmic thinking skills that you hone in this class are applicable to virtually any future. The SAAST course in computer science is designed for students with no significant prior programming experience. If you already have significant programming experience then we recommend looking into our other SAAST offerings, e.g., robotics and computer graphics.

### **Staff**

Instructor Peter-Michael Osera ([posera@cis.upenn.edu](mailto:posera@cis.upenn.edu))  
Teaching assistants TBA

### **Technology**

For this course, we will be using the Python programming language. Python (<http://python.org/>) is a lightweight, high-level programming language suitable for both learning programming fundamentals and quick-and-dirty software development. Students will have access to UPenn computing resources and labs to complete their assigned work. Students may also bring their own laptops to class and the course staff will assist in helping them set up the necessary programs so that they can write Python programs on their own machines.

### **Structure**

The only way to learn computer programming is to do it. A lot. And the structure of the course reflects this ideal. Each day consists of lecture introducing the programming concept(s) of the day followed by labs run by our teaching staff that will allow you to practice those concepts. The assigned homework drills deeply into these concepts further.

By the end of the course, the class will work together to create a shared final project. Everyone will add a component to this project and present their contribution on the last day of class.

### **Grading**

There are no exams in this class. You will be graded on your lab work and homework as well as your final presentation.