

Course: Programming 1, CS 21 SMC

Time: MWF 10:20-11:20 a.m. @ GAV 250

Text: Python Programming: An Introduction to Computer Science by John Zelle

Instructor: Adam Lucas (office Gal 103D)

Office Hours: MF 11:30-12:30, and 2:15-2:45, or by appointment

Contact info: Adam SMC: GAL 103D
Adam Home: 415-885-1680
arl3@stmarys-ca.edu

Prerequisite: One year each of high school algebra I, II and geometry or equivalent. No prior programming experience is necessary but students are expected to have a **strong** interest in programming.

Course Description: This class will be an introduction to computer programming in Python. The primary goal of the class is to teach you problem solving, design and programming. One of the reasons we learn Python as a first programming language is because of its simple syntax. For example, here is a comparison of what students need to type to print “hello world” on the computer screen in C++ and Python:

In C++:

```
#include <iostream>
using namespace std;

int main()
{
    cout << "hello world\n";
    return 0;
}
```

In Python:

```
print "hello world"
```

As you can see, the syntax for Python is much simpler than for C++. Because of Python’s simple syntax we will quickly be able to design some cool graphics like widgets and dice poker. Besides being fun, graphics illustrates the concept of object-oriented design (OOD) very well.

To prepare you for Programming 2, we will devote the last three weeks of class learning C++. I believe that your knowledge of Python will make learning C++ much easier.

Homework: There will be nightly programming assignments as well as nightly reading assignments. The average student will need to spend 3 hours per class on homework. You may discuss the homework by forming a group and studying with your peers. Late homework will **not** be accepted.

Each homework problem is worth a maximum of 10 pts:

Functionality 0-7 pts: This part of the grade is based purely on the behavior of the program. If it does not work in any useful way, then it gets 0 for functionality. I do this since it shows that functionality is a spectrum, and it shows that it is possible to build up functionality incrementally.

Style 0-3 pts: I remove one point for any egregious style violation, such as limited or no comments, very bad variable names, not factoring code, using global variables.

Academic Honesty: The Saint Mary's policies regarding academic honesty detailed in the student handbook apply to this course. I encourage you to work with other students on the homework, but your write-up of the homework should be mostly on your own. Where your programs depends heavily on another's work, say so.

Attendance: Attendance is required and roll will be taken at the *beginning* of each hour. If you are not in your seat when roll is taken, you may be considered absent, so be on time. You are allowed to miss **three** classes without affecting your grade. After your grade is dropped one step (A- to B+, C+ to C, etc.) for each two successive misses. SMC athletes are excused to attend team commitments but are responsible for notifying me by September 15 of all athletic conflicts.

Exams/Projects: There will be two 1-hour tests, a project and a 2-hour cumulative final. Missed examinations are given an automatic grade of F. The only exceptions to this, *made at the discretion of the instructor*, are in cases of major emergencies. Contact me ASAP if you think you are going to miss an exam.

Exam/Projects Dates: Wednesday, September 19 (in class) 1 hour exam
Wednesday, October 24 (in class) 1 hour exam
Monday December 3: 11:30am Project due
Monday December 3 11:30-1:30, 2 hour final exam

Grading: Your final grade is computed as follows:
Homework 25%
WeBWork/ i-clicker score/ final project 15%
Test 1: 15%
Test 2: 15%
Comprehensive final exam 30%

Tentative Schedule:

Week 1 (Aug 27) Chap 1,2,3: Simple programs, numeric data types
Week 2 (Sept 3) Chap 4, start Chap 5: String data type, IO, file processing, objects
Week 3 (Sept 10) Chap 5: Intro to objects and methods via graphics
Week 4 (Sept 17) Review **Test 1**, Chap 6: functions
Week 5 (Sept 24) Chap 7,8: Decision structures, loops and booleans
Week 6 (Oct 1) Chap 9: Simulation and design: simulate racket-ball game, top down design, bottom-up implementation
Week 7 (Oct 8) Chap 10: Defining classes, example widgets
Week 8 (Oct 15) Chap 11: Data collection ,lists
Week 9 (Oct 22) Review, **Test 2**
Week 10 (Oct 29) Chap 12: Object oriented design (OOD)—racket ball simulation
Week 11 (Nov 5) Chap 12 cont: OOD—Dice Poker graphical simulation
Week 12 (Nov 12) Intro C++ for Python programmers handout: C++ vs Python, Data types in C, IO, compiling, expressions and operator precedence, decision statements
Week 13 (Nov 19) Intro C++ for Python programmer handout: functions, header files
Week 14 (Nov 26) Intro C++ for Python programmer handout: recursion, **OOD projects due**, review