Faith Christian Academy: Computer Programming

Syllabus Updated 2013-05-22

**Course Description:** Create software using the Python scripting language. Learn coding and concepts for command line programs, windows programs, 2D graphics, and 3D graphics.

**Instructor**: Jake Gustafson, 215-257-4577 x3328, [jgustafson@my-fca.com](mailto:jgustafson@my-fca.com)

**Materials Used**: “Computer Programming ” written by Jake Gustafson (19 yrs. Total programming experience including QBASIC, Visual Basic, SQL, Access VBA, C++ [Win32 API, MFC, .NET, DirectX, SDL], C#, and Python); Example code & program footage from various sources; Public domain and free-licensed imagery & objects for use in projects.

**Software Used**: Python project official tools (runtimes & IDLE for command line applications), wxGlade (windows programming), pygame (2D graphics and other direct media), and Panda3D (3D graphics engine developed and used by Carnegie Mellon University and Disney). Python is a relatively new language that is becoming more widely used in jobs in computer science, web development, and can be used in advanced video editing and 3D graphics (for task-specific scripting and plugin development).

**Goals**: Inspire interest in computer science, give students ability to solve problems using their own software, give students a deeper understanding of computers from learning how their interface is designed, prepare students for software and web design jobs that require or can make use of python (python is actually often required by web design or software jobs but finding training in Python can be difficult to find).

**Grading**: Regular assignments for each part are worth 10 points, last about one week, and generally involve mastering one or two methods in a program that would require them (some statements are often used in later programs as well). Creative projects, which require a combination of methods to accomplish a larger task which may require finding photos or other data (& sometimes two weeks) are worth 20pts. Each quarter has about 100pts, so each project is worth about 10% or 20% of the grade for that quarter. Each quarter also has a 10pts for participation.

|  |  |  |  |
| --- | --- | --- | --- |
| **Point Value** | **Part & Name** | **Due** | **Quarter** |
| 10 | Console 1: “print” and variables | 8/31 | 1 |
| 10 | Console 2: “raw\_input” statement, literals, and errors | 9/7 | 1 |
| 10 | Console 3: combining strings, and Python indentation | 9/14 | 1 |
| 10 | Console 4: booleans, “while” statement, and nested statements | 9/21 | 1 |
| 10 | Console 5: “list”, “for”, and “range” statements | 9/28 | 1 |
| 10 | Console 6: design, logic - flowchart project | 10/5 | 1 |
| 20 | Console 7: converting ideas to code, writing code, testing | 10/12 | 1 |
| 10 | Console 8: classes, random, review of types | 10/19 | 1 |
| 20 | Console 9: write file and read file | 10/26 | 1 |
| 10 | Console 10: exception handling | 11/7 | 2 |
| 10 | GUI 1: Add a box layout to create a row of buttons. | 11/15 | 2 |
| 10 | GUI 2: Add an event to a button. | 11/22 | 2 |
| 10 | GUI 3: Create a dialog box; make a button show it. | 11/29 | 2 |
| 10 | GUI 4: Using Variables in a GUI—button to let the user buy virtual applesauce! | 12/6 | 2 |
| 20 | GUI 5: Create tree chopping program without teacher help. | 12/13 | 2 |
| 20 | GUI 6: Make buttons load images into a widget. | 12/20 | 2 |
| 10 | GUI 7: ListBox | 1/8 | 2 |
| 10 | GUI 8: Model-View-Controller | 1/17 | 2 |
| 10 | GUI debugging challenge - debug an existing GUI program | 1/17 | 1 |
| 10 | Canvas 1: moving box | 1/28 | 3 |
| 10 | Canvas 2: drawing graphics manually | 2/6 | 3 |
| 10 | Canvas 3: Main Event Loop |  |  |
| 20 | Unit 3 Quiz: Input and Output | 2/6 | 3 |
| 10 | Canvas : Making sprites interactive (pushing sprites) | 2/15 | 3 |
| 5 (participation) | Videos about the software industry (watch, ask, answer) | 2/22 | 3 |
| 10 | Canvas : Making your own sprite and loading it in your program | 3/6 | 3 |
| 20 | Unit 4 Quiz: 3D Graphics Terms | 4/8 | 4 |
| 10 | OpenGL 1: creating your first 3D program | 4/24 | 4 |
| 10 | OpenGL 2: 3D position, rotation, and camera | 5/3 | 4 |
| 10 | OpenGL 3: terrain | 5/17 | 4 |
| 20 | OpenGL 4: (Final Project) Interactive 3D | 6/6 | 4 |