Introduction to Programming

COURSE SYLLABUS

Lafayette High School
Mr. Derrick Smith
401 Reed Lane
Lexington, KY 40503
(859) 381-3474
derrick.smith@fayette.kyschools.us
http://www.derricksmithweb.com

Course Description

This is an introductory course offering a very broad perspective on programming, with more specific instruction using Python, Java, Scratch and possibly a few other languages as time permits. The goal is to introduce what a computer programmer does and how one uses various programming tools to accomplish the objectives. Emphasis will be placed on creativity, and commonalities between languages such as logic, flow control, loops, arrays, objects, etc.

Materials

You will need the following supplies for this course:

- Folder
- Pen or Pencil
- USB flash drive

Grading Scale

A	92-100
B	83-91
C	74-82
D	65-73
F	64 or below

Evaluation Procedure

Your grade will consist of the following 3 categories:

1. Category 1 Formative Assessments (40%)
    a. Daily Work
2. Category 2 Summative Assessments (45%)
    a. Quizzes
    b. Tests
    c. Projects
3. Category 3 Final Exam (15%)

Make-up Work

Make-up work will be accepted at any time prior to the start of final exams. Once final exams have started, make-up work will no longer be accepted. THERE ARE NO EXCEPTIONS TO THIS POLICY. If you are absent, you can track all assignments on my website at http://www.derricksmithweb.com. It is your responsibility to make up any missing assignments.

Missing Assignments

Missing assignments that are turned in will be graded. **However, grading missing assignments is a low priority. Most missing assignments will be graded within one week of turning in.**

Homework

Homework will rarely be given. However, each student is expected to complete work in class during each and every class period. The SAM training program can be accessed from most home computers, so if you miss class one day, you can likely work from home.

Course Topics

1. Demonstrate knowledge of the program development life cycle.
2. Design, develop, compile, debug, test, run, and document programs in the language studied.
3. Design and develop programs using operators and assignments.
4. Design and develop programs that properly use variable, constants, data types, and objects.
5. Design and develop programs that use sequence, selection, and repetition structures.
6. Design and develop programs that use simple data structures.
7. Design and develop programs that use effective error and exception handling.
8. Design and develop programs that implement user-defined methods and modular programming.
9. Design and develop programs that implement file processing.
10. Design and develop programs that implement fundamental features that are unique to the language studied.
11. Design and develop programs using object oriented programming features, if applicable to the language studied.
12. Evaluate and critique effectiveness and efficiency of code written.