Syllabus: Computer Programming

A. General Information

Units	1
Prerequisites	Required: Algebra, Applications or demonstrable computer skills Helpful: Geometry, Trigonometry
Instructor	William B. Smith
Textbook	Required: none (online text) Optional: True BASIC by Problem Solving (Hahn); True BASIC Bronze Edition (Kemeny and Kurtz); Alice (http://www.alice.org/); Learning to Program with Alice, 2nd ed. (Dann, et.al.)
Materials	Pen, paper, journal (composition book),
Standards	A >= 93%, B >= 83%, C >= 70%, D >= 60%
Required Class Hours	Five 41-minute days per week
Open Lab Hours with Instructor Present	2:17 PM to 3 PM schedule permitting or by appointment

B. Grading

ITEM	POINTS	COMMENTS
Program Projects	1	per program, half credit also given
Exams	4	based on 4.0 scale
Online Midterm Exam	10%	of final grade, skill-based program included
Online Final Exam	10%	of final grade, skill-based program included

C. Week by Week

WEEK	DESCRIPTIONS
1	Getting Started with Alice
2	Program Design and Implementation
3	Simple Control Structures
4 - 5	Classes, Objects, Methods, and Parameters
6 – 7	Events
8 - 9	Functions
10 - 11	Loops
12 - 13	Recursion
14 - 15	Lists
16 - 17	Inheritance
18	Review and Midterms
19	Getting Started: Introduction to BASIC Programming
20	Output
21	Math Operations and Variables
22	Input
23	Formatting Output
24	FOR-NEXT Loops
25	DO Loops
26 - 27	IF-THEN Statements
28	Predefined Functions
29	The Special Case of Random
30	Predefined Functions
31	SELECT CASE
32	READ-DATA
33	Arrays
34	Matrices
35	Graphics and Sound
36	Review and Final

D. Benchmark Questions and Skills

WEEK	BENCHMARKS
1	Can all students navigate through the Alice interface and complete the tutorial?
2	Can all students identify and create a scenario?
3	Can all students identify instructions, control structures, functions, and expressions? Can all students design and implement a simple program?
4 - 5	Can all students create primitive, class-level, and world-level methods utilizing stepwise refinement?
6 - 7	Can all students create interactive programs that respond to events?
8 - 9	Can all students manipulate execution control with If/Else and Boolean functions? Can all students generate random numbers and random motion?
10 - 11	Can all students recognize and use simple loops, infinite loops, and conditional loops in their programs?
12	Can all students incorporate recursion into programs?
13 - 14	Can all students organize objects or information of the same type into lists?
15 - 16	Can all students create and invoke class-level variables? Can all students manipulate an array?
17 - 18	Given the problem by the instructor, can all students demonstrate a proficiency in Alice programming by writing a program with a decision to solve the problem? Given the problem by the instructor, can all students write a program that reads to and writes from an array?
19	Can all students recognize the parts of a computer program? Can the students read and write simple programs?
20	Can all students send output to the screen or a printer? Can they print series of numbers or characters at different places on the screen, and in different orders?
21	Can all students combine operands and operators in correct sequences? Can all student recognize string and numeric variables? Can all students use variables to store data?
22	Can all students alter programs during execution? Can all students write programs to allow data input that is stored in variables, manipulated and output to the screen?
23	Can all students format text and numbers?
24 - 25	Can all students identify types of loops and when the loops are used? Can all students use pretest iterations?
26	Can all students transpose FOR-NEXT loops into DO loops? Can all students define a sentinel and where it is used?
27 - 28	Can all students describe and use the conditional branch? Can all students use decisions to vary output?
29	Can all students list predefined functions, determine output for the functions, and write programs with embedded functions?
30	Can all students vary random output? Can all students create a menu?
31	Can all students read data into a program? Given different examples, can all students locate errors in read and data statements?
32	Can all students dimension, fill, and read arrays?
33 - 34	Can all students create a matrix? Can all students sort numeric and string matrices? Can all students write a program to search for a matrix element?
35 - 36	Can all students define resolution? Can all students draw objects, change the objects' colors, and move the objects? Can all students combine graphics with sound in a program?