Swift Syllabus Schedule

Wednesday	**Thursday**	**Homework Due Next Wednesday**
Sept. 14: *Learn:* Swift environment setup, REPL demonstration, *Do:* worksheet on creating constants, their types, and print(). • Presentation 1 • Worksheet 1	15: *Learn:* constants vs. variables, arithmetic, type inference, string interpolation. *Do:* group programming assignment involving variables, computation, and printing. • Presentation 2 • Assignment 1	Write-Up 1
21: *Learn:* Boolean expressions, If statements, while loops, *Do:* worksheet on boolean expression, if statements, while statements. • Presentation 3 • Worksheet 2	22: *Do:* group programming assignment involving if statements and while loops. • Assignment 2	Write-Up 2
28: *Learn:* Arrays, indexing, iterating arrays using while loops, *Do:* worksheet with arrays and while loops. • Presentation 4 • Worksheet 3	29: *Learn:* For loops, *Do:* worksheet with for loop practice, group programming assignment with arrays and loops. • Presentation 5 • Worksheet 4 • Assignment 3	Write-Up 3
Oct. 5: *Do:* **TEST**, *Learn:* Functions (not on test). *Do:* worksheet with functions. • Test 1 • Presentation 6 • Worksheet 5	6: *Learn:* more functions, *Do:* programming group assignment with functions. • Presentation 7 • Assignment 4	Write-Up 4
12: *Learn:* Basic classes, initializers, instance variables, methods, *Do:* worksheet on basic classes. • Presentation 8 • Worksheet 6	13: *Learn:* Basic subclassing and polymorphism, overriding, final project, *Do:* group programming assignment with some classes and subclasses. • Presentation 9 • Presentation 10	Final Project Part 1 (see below)

19: *Learn:* Overriding review, access modifies, enums, *Do:* worksheet with access modifier and enums. • Presentation 11	Work time for final project	Final Project Part 2
Work time for final project	Work time for final project	Final Project Part 3
Work time for final project	Work time for final project	Final Project Part 4
Work time for final project	Work time for final project	

Learning:

Learn segments consist, for the most part, first of a presentation, with slides that will be emailed out for reference, followed by live coding demonstration.

Activities:

Worksheets consist of both reading segments of code and trying to understand meaning and execution, and producing new code. Group programming assignments will roughly come in 3 types: first, students will be expected to take existing code that performs some operation, understand it, and modify it to do what they want. Next, students will be expected to produce code from scratch. Finally, students will be expected to find bugs in provided code.

Write-Ups:

Each write-up consists of several individual programming problems. Students are expected to complete each problem, with both complete solutions and proper justification for their code via comments. Each successive write-up *will* be cumulative in knowledge: as in, later write-ups will require knowledge of topics of previous weeks. However, write-ups will *not* build upon the actual code of past write-ups (unlike projects, see below). In addition, write-ups are individual work only. On Write-Ups, students will be expected primarily to produce substantial amounts of original code from scratch.

Projects:

Each project builds upon the code base created from the previous project. In general, the goal of the projects system is to build up a fairly substantial code base, and in doing so motivate object-oriented programming concepts. This can be done either individually or with 1 partner. However, if done as a pair, each person must be able to explain every part of the code.