Appendix P ….. Time Allocation for Lessons and Tests

Lesson	Time	Comments
"First day" activities	1 day	Pass out books, demonstrate how to log-on, create project folder, learn how to launch and configure programming environment.
Lesson 1	1 day	Enter the "Hello World" program into the computer and execute.
Lesson 2	1 day	Illustrate each point of Lesson 2 by modifying the code of the "Hello World" program of Lesson 1. Assign the exercise on Lesson 2 as homework. There should be some time to work on this towards the end of the period and time to check answers.
Lesson 3	2 days	Illustrate each point of Lesson 3 by running code. Assign Exercise on Lesson 3. Grade assignment at end of 2nd day.
Test through Lesson 3	1 day	Allow the students to work on the test, take it up at the end of the class, and let them know when them come back the next day they can make any corrections they look up that night. Keep beginning students from becoming discouraged from the start by making this an open-book test.
Lesson 4	2 days	Have the students run many of the code examples in their IDE (integrated development environment) Some of the problems on the exercise must be finished as homework in order to fit this lesson into one day.
Lesson 5	1 day	Run several of the code examples.
Lesson 6	1 day	Run several of the code examples.
Lesson 7	1 day	Do the first project together in class and assign the "Full Name" project as a written assignment. Typical grades range from 92 to 51. Problems 5, 9, 11, 12, 2, and 25 were those most often missed. On the day after the test go over these specific problems.
Test through Lesson 7	1 day	At this point still let the students use the book for the test… try to build up their confidence.
Lesson 8	2 days	Many of the code example need to be run on the computer as they are discussed.
Lesson 9	2 days	Be sure to actually run the first two code examples.
Lesson 10	3 days	Definitely run the code "menu" example.
Test through Lesson 10	1 day	This test may be difficult for some. On the day before the test let the students look over a copy of the test for about 10 minutes so there will be no surprises on the day of the test.
Lesson 11	3 days	This is the most important lesson so far. Be sure to run several of the code examples. This is where we begin to acclimate the students to "contest type problems".
Lesson 12	3 days	Again, run many of the code examples… very important concepts here. After the 14 regular exercise problems are completed, give the 5 "contest type" problems as a quiz.
Lesson 13	2 days	Emphasize the techniques for storing a char into a *String* and vice versa. Have students memorize the ranges of ASCII codes.
Lesson 14	2 days	Use chalk board for demo of conversion techniques.
Test through Lesson 14	2 days	This will be a lengthy test and it is suggested that it not be an open-book test. Many students will need two days. Let the first day be an eye-opener for them so they will study overnight and continue the next day.
Make-up test through Lesson 14 (**Alternate Test through Lesson 14**)	2 days	In order not to discourage students who do poorly on the original test, you might possibly want to give this 16-question re-take. Spend one day correcting the mistakes from the original and then one day taking this new test… The questions on this test are mostly what is likely to have been missed on the original test.
Lesson 15	3 days	Spend time on this lesson! This is the most important lesson so far. Have students enter and test the code for the *Circle* class. There are 20 questions on the exercise for this lesson. If the students do poorly on the exercise there is a "redemptive" quiz that could be given.
Lesson 16	3 days	This is a follow up to lesson 15…very important concepts here.

Test through Lesson 16	1 day	This is primarily a test on objects and classes (lesson 15 and 16)
Lesson 17	3 days	Students will find this much easier than the previous lessons on objects.
Lesson 18	3 days	Let students know that arrays will be used in nearly all future lessons.
Lesson 19	4 days	Be sure to do the programming projects.
Test through Lesson 19	2 days	It is suggested that this test be split across two days. Overnight they can study what they saw on the test and didn't understand.
Lesson 20	2 days	
Lesson 21	1 day	
Lesson 22	1 day	
Test through Lesson 22	2 days	This is a difficult test. Let the students work on it for 30 minutes or so the first day, take it up, let them study overnight, and then finish the second day.
Lesson 23	4 days	Plan to spend 1 day going over the material in the textbook. The second day can be devoted to doing and explaining the exercises. The programming project will also take a complete period if the students are forced to do most of it themselves. The contest type problems will require a day. Some of those problems are tricky and will require some explanation.
Lesson 24	3 days	Many lesson from this point on depend on inputting data from a file. Make sure the students get a good foundation.
Test through Lesson 24	1 day	This test is considerably shorter than the others and probably easier.
Lesson 25	3 days	Honing skills with file input.
Lesson 26	1 day	Actually this lesson can be done in half a period.
Lesson 27	2 days	Be sure students keep the *BaseClass* class. They will paste code from it into many of their future projects.
Test through Lesson 27	1 day	
Lesson 28	3 days	The project in this lesson will take an entire day for most students.
Lesson 29	2 days	
Test through 29	1 day	
Lesson 30	3 days	Be sure to do the Monte Carlo project.
Lesson 31	1 day	Stress the *append* and *toString* methods.
Lesson 32	3 days	DeMorgan's theorem is very important.
Lesson 33	1 day	
Test through 33	1 day	
Lesson 34	2 days	Some important concepts are here.
Lesson 35	3 days	
Test through 35	1 day	
Lesson 36	3 days	This can be done in three days; however, this is such an important lesson that it might be more desirable to allocate 4 days.
Test through 36	1 day	This test focuses strictly on Lesson 36, the inheritance lesson.
Lesson 37	2 days	
Lesson 38	2 days	
Test through 38	1 day	
Lesson 39	2 days	
Lesson 40	3 days	This lesson on recursion is especially important.
Test through 40	1 day	
Lesson 41	6 days	Spend one day for each sorting type.
Test on Lesson 41	1 day	
Lesson 42	1 day	
Lesson 43	3 days	ArrayList. Spend at least one day on the project.
Lesson 44	4 days	Iterators
Test on Lesson 44	1 day	
Lesson 45	3 days	These concepts are very important. Be sure to do all three projects.
Test on Lesson 45	1 day	
Lesson 46	2 days	
Lesson 47	3 days	
Test on Lesson 47	1 day	
Lesson 48	2 days	
Lesson 49	3 days	

Lesson 50	2 days	
Test on Lesson 50	1 day	
Lesson 51	3 days	
Lesson 52	3 days	
Test on Lesson 52	1 day	
Lesson 53	2 days	
Lesson 54	2 days	
Test on Lesson 54	1 day	
Lesson 55	3 days	
Lesson 56	3 days	
Lesson 57	3 days	
Test on Lesson 57	1 day	