Study Questions
CSC 207, “Algorithms and Object-Oriented Design”
Department of Computer Science
Grinnell College

For August 30, 2019

1. What language construct in Java is most similar to a function definition in C? What are the most significant differences between C function definitions and their Java analogues?

2. What does the keyword static signify when it occurs in a field declaration or in the signature of a method definition?

3. Examine the short Java program at
   /home/hajiamini/courses/CSC207/code/HelloWorld.java
   Why do some of the comments in that program begin with /** rather than /*?

4. How would you find out what methods the objects of the Java String class support and how to invoke those methods?

For September 2, 2019

5. What is the difference between Java’s two right-shift operators, >> and >>>? Give an example in which applying these operators to the same operands would yield different results.

6. Do the lab Starting Out in Java. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For September 4, 2019

7. What’s the difference between a reference and a pointer? In what contexts, in Java, does a reference variable behave like a C pointer?

8. Java has no operators analogous to the dereferencing operators in C (the unary * operator and the -> operator, which dereferences its left operand and selects a field from the result). How does Java handle dereferencing?

9. Do the lab References and Objects. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For September 6, 2019

10. What’s the point of information hiding and encapsulation? How are these ideas related to abstraction in the object-oriented programming?

11. The author’s definition of the BigRational class ensures that BigRational objects will always satisfy certain invariants. What are they?

12. Do the lab Objects and Classes. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For September 9, 2019

13. How can one direct the Java compiler to look for importable packages in particular directories?

14. Do the lab Java Standard Libraries omitting the optional exercise at the end. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.
Study Questions — page 2

For September 11, 2019

15. What are the preconditions that the java virtual machine must satisfy before it can start to execute the main method of the class specified on the command line?

16. Do the lab The Java Virtual Machine. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way and your answers to the questions posed.

For September 13, 2019

17. What advantage does Java’s immutable String type have in comparison with the mutable strings of C and Scheme?

18. Do the lab Strings and StringBuilders. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

19. The lab prompts you to use StringBuilder objects when assembling strings incrementally. Explain why this is more efficient than working with String objects and concatenating them together.

For September 16, 2019

20. What happens if an exception is thrown during the execution of two different try-blocks? (This can happen if a try-block includes a call to a method that contains another try-block, which includes a call to a method in which the exception is thrown.)

21. What happens if no try-blocks are being executed when an exception is thrown?

22. Do the lab Exceptions. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For September 18, 2019

23. Do the lab The Eclipse Integrated Development Environment. (You may omit the section “First Encounter” and exercises 1–4 if you have used Eclipse before.) Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For September 20, 2019

24. The textbook claims (page 53) that Java lacks the formatting options that C and C++ provide. Read the description of the format method in the java.lang.String class and compare the facilities it provides with those of the printf function in C.

25. Do parts 1–3 of the lab Input and Output. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For September 23, 2019

26. What does it mean for one class (A) to be “type-compatible” with another class (B)?

27. What does the (implicitly defined) super method do, and under what conditions can it be used?

28. Do the lab Inheritance. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.
For September 25, 2019

29. Do the lab Generics. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For September 27, 2019

30. What is the purpose of a version-control system? Given this purpose, what kinds of files should not be placed under version control?

31. One of the distinguishing features of the Git version-control software is that there need not be any central master repository for a project. Under what circumstances would this be an advantage?

32. Do exercises 1 through 9 in the The Git Version-Control System. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce output either from a Java program or from Git itself.

For September 30, 2019

33. Complete the lab The Git Version-Control System. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce output either from a Java program or from Git itself.

For October 2, 2019

34. Why doesn’t Java allow a class to extend two or more parent classes?

35. If Foo is an interface, why is it (a) permissible to declare a variable or parameter of type Foo but (b) not permissible to write new Foo() to create a new object that implements Foo?

36. Do the lab Polymorphism and Interfaces. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For October 4, 2019

37. Why might a programmer want to resize an array at run time?

38. Under what conditions can an array of elements of object type A be assigned to a variable that is declared to be an array of elements of object type B?

39. Do the lab Population Density of Iowa Counties. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For October 7, 2019

40. What does it mean to say that a get operation on an ArrayList “is O(1)”?

41. Weiss argues that, in evaluating the running time of an operation that adds an element at the end of an ArrayList, we can ignore the occasional need to resize the underlying array because it happens so infrequently. Similarly, the Java API documentation says that the add method “runs in amortized constant time.” But resizing is very time-consuming when it does happen, and it entails copying every element in the array. How can these seemingly contradictory observations be reconciled?

42. Write an “enhanced for-loop” (as described in section 2.4.6 of the textbook) to compute the sum of the elements of a LinkedList of Integer values.
For October 9, 2019

43. Do the lab Assertions and Loop Invariants. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For October 14, 2019

44. Why did the designers of the Java libraries arrange for the LinkedList class to implement the Queue interface while the ArrayList class does not?

45. The textbook uses the names enqueue, dequeue, and getFront for the operations of adding an element at the rear of a queue, removing an element from the front of a non-empty queue, and examining the element at the front of a non-empty queue. What names does the Queue interface use for these operations?

46. Do the lab Stacks. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For October 16, 2019

47. Do exercises 1 through 6 of the lab Mergesort. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For October 18, 2019

48. Do the lab Quicksort. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For October 28, 2019

49. What are the primitive operations on priority queues? How do they differ from the primitive operations on queues?

50. Do the lab Priority Queues. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For October 30, 2019

51. What are the advantages and disadvantages of using header nodes in the implementation of linked lists?

52. In the implementation of the Java standard LinkedList API in section 17.5 of the textbook, why is the nested class Node declared static while the LinkedListIterator class is not?

53. Do the exercises 1 through 22 of the lab Circular Lists. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For November 1, 2019

54. Complete any exercises from the lab Circular Lists that you have not already done. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

55. How can one arrange for special formatting, font effects, etc. to appear in the documentation that javadoc produces?
56. Find one of the class definitions that you have written for an earlier lab. Add javadoc comments to it, save the revised version, and run javadoc to have it construct the documentation. Review the resulting HTML pages and revise the javadoc comments to make them look even better or to make them more useful to an application programmer who might want to use the class you defined.

For November 2, 2019

57. Why might one want to override the equals method in a class in an application in which objects of that class are going to be put into sets or used as keys in maps?

58. Do the lab Maps. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For November 6, 2019

59. Read the GNU/Linux manual page for the urandom device by opening a terminal window and typing man 4 urandom at the prompt. Then explain why it is better to use a linear-congruential pseudo-random-number generator than a random-number generator that pulls values from /dev/urandom in a research simulation where exact reproducibility of results is required.

60. Do exercises 1 and 2 in the lab Random-Number Generation using /dev/urandom. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For November 8, 2019

61. In Weiss’s method for permuting the elements of an array (Figure 9.7, page 405), each element a[j] of the array is swapped, once, with an element in a position randomly selected from the range from 0 to j (inclusive). Why is this better than swapping it with an element randomly selected from the entire array (by choosing the position from the range from 0 to a.length – 1)?

62. Do exercises 3 through 10 in the lab Random-Number Generation using /dev/urandom. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For November 11, 2019

63. What invariant do binary search trees observe (beyond the basic structure common to all binary trees)?

64. Do the lab Trees and Traversals. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For November 13, 2019

65. If we form a binary search tree from a set of integers by alternately inserting the least of the remaining items and the greatest of the remaining items, will the resulting binary search tree be bushy or stringy? Justify your answer.

66. Do the lab Implementing Treesort. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.
Study Questions — page 6

For November 15, 2018

67. In addition to the binary search tree order property, what invariant do AVL trees satisfy?

68. In addition to the binary search tree order property, what invariants do red-black trees satisfy?

69. Weiss considers two possible strategies for insertion into a red-black tree ("bottom-up insertion" in section 19.5.1 and "top-down insertion" in section 19.5.2). Explain why he prefers the latter strategy, even though either one would yield a correct result.

For November 18, 2019

70. State the invariants for aa trees (besides the binary search tree ordering property). Explain how skew and split operations restore the aa trees invariants.

For November 20, 2019

71. In Weiss’s implementation of the TreeMap class, why does the keySet method return a view of the underlying class rather than a freshly allocated set?

72. Do the lab Extending TreeSet. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For November 22, 2019

73. What data structure is used in the implementation of hash tables?

74. If two or more objects have the same hash code, what will happen if both are added to the same HashSet?

For November 25, 2019

75. Why would programmers be reluctant to use the “implicit” array representation for binary search trees as well as for heaps?

For November 29, 2019

76. What is an adjacency matrix? What are the advantages and disadvantages of using an adjacency matrix to represent the arcs of a directed graph, as compared to a representation using adjacency lists (as in Weiss’s implementation of the Graph class)?

77. The main method in Weiss’s implementation of the Graph class reads in a description of a graph from a text file, with one line of text per arc. The line contains the name of the source vertex, the name of the target vertex, and the arc weight (as the numeral for an integer value).

(a) Weiss provides no way to read in an unlabelled graph. What changes would one have to make in the data structures or in the input routines in order to accommodate such graphs?

(b) Weiss provides no way to read in a graph containing isolated vertices, that is, vertices that are neither the source nor the target of any arcs. What changes would one have to make in the data structures or in the input routines in order to accommodate such graphs?

(c) Weiss provides no way to read in graphs with arc weights that are real numbers but not integers. What changes would one have to make in the data structures or in the input routines in order to accommodate such graphs?

For December 2, 2019

78. How does Weiss’s Java implementation of the breadth-first solution to the shortest-path problem on unweighted graphs distinguish the vertices that have been visited from those that have not, so that the former are never reinserted into the queue?
79. Do exercises 1 through 7 of the lab *Word Ladders*. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For December 4, 2019

80. How is Dijkstra’s algorithm different from the breadth-first search algorithm in terms of finding the shortest path solution for a vertex? How does the Dijkstra’s algorithm implement that solution?

81. Do the lab *Dijkstra’s Algorithm*. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.

For December 6, 2019

82. In the unweighted-graph algorithm for finding shortest paths in a graph, the key invariant was that when a vertex emerged from the queue, its dist and prev fields were final and reflected the shortest paths from the starting vertex. In Dijkstra’s algorithm, we had the same invariant (except with a priority queue) and the additional invariant that the dist and prev fields of the items adjacent to the vertex emerging from the queue reflected the shortest paths from the starting vertex using only vertices that had already emerged from the queue as intermediate path elements. What is the corresponding invariant for the Bellman-Ford algorithm?

83. What does the scratch variable at line 18 of Figure 14.29 (the Bellman-Ford algorithm) represent when the algorithm throws an exception at line 19?

For December 9, 2019

84. As a function of the number of vertices and edges in an acyclic directed graph, what is the running time of the topological sorting algorithm?

For December 11, 2019

85. What is the “equivalence problem” that the disjoint-set class is supposed to solve?

86. For finding a minimum spanning tree from a weighted, undirected graph, describe (a) the equivalence relation between some elements of the application, (b) the equivalence classes induced by that relation, and (c) how a solution to the minimum spanning tree is constructed by calls to the find and union methods.

87. Suggest a practical application in which it would be useful to compute a minimum spanning tree for a given graph.

88. Do exercises 1-4 in the lab *Minimum Spanning Trees*. Write a short summary confirming that you completed each step and reporting any difficulties you encountered along the way. Include any Java code that you wrote and report the results of all exercises that produce program output.