Study Questions
CSC 207, “Algorithms and Object-Oriented Design”
Department of Computer Science
Grinnell College
revised December 13, 2018

For August 31, 2018

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/reseda/object-oriented-programming/code/HelloWorld.java
   Why do some of the comments in that program begin with /** rather than /*?

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

For September 3, 2018

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 5, 2018

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 7, 2018

10. What’s the point of information hiding and encapsulation? How are these ideas related to the ideas of modularity and abstract data types that we encountered in CSC 151 and CSC 161?

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 10, 2018

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.

For September 12, 2018

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 14, 2018

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

18. Do the lab “Strings and StringBuilders,” 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.

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 17, 2018

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 19, 2018

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 21, 2018

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 24, 2018

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 26, 2018

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 28, 2018

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 October 1, 2018

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 3, 2018

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 5, 2018

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 8, 2018

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 10, 2018

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 12, 2018

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 15, 2018

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 17, 2018

48. Do exercise 8.11 on page 386 of the textbook.


50. 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 19, 2018

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

52. 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 29, 2018

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

54. 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?

55. 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 October 31, 2018

56. 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.

57. How can one arrange for special formatting, font effects, etc. to appear in the documentation that javadoc produces?

58. 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, 2018

59. What are the advantages and disadvantages of programming with “views,” such as those described in section 6.10 of the text?

60. 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?

61. 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 5, 2018

62. 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.

63. 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 7, 2018

64. 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)?

65. 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 9, 2018

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

67. In the BinarySearchTreeWithRank class, if a call to the insert or remove method fails when the node at the end of a branch is reached, the size fields of the nodes along that branch
should not be modified. How does Weiss’s implementation prevent the execution of the `size++`; or `size--;` statement in each of the recursive calls?

68. 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 12, 2018

69. 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.

70. 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.

For November 14, 2018

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

72. How do we know that a tree that satisfies the AVL invariant is bushy enough to ensure that the worst-case running time of all of the `BinarySearchTree` methods is \(O(\log N)\)?

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

74. 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 16, 2018

75. State the invariants for aa trees (besides the binary search tree ordering property) and explain how they are restored before the end of an `insert` operation.

76. Why are the `insert` and `remove` operations on aa trees implemented recursively rather than iteratively?

For November 19, 2018

77. 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?

78. Do the lab “Extending TreeSets.” 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 21, 2018

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

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

81. Why can a hash-table implementation that uses separate chaining to resolve collisions tolerate a larger load factor before requiring rehashing than an implementation that uses linear probing?

For November 26, 2018

82. Why would programmers be reluctant to use the “implicit” array representation for binary search trees as well as for heaps?
83. Suppose that you are performing an external sort using a total of four external tape drives, one of which contains the original unsorted data. You propose to use a polyphase multiway merge in which three runs, one on each input tape, are merged into a single run, which is placed on the output tape. When any of the input tapes contains no more runs, it will be rewound to serve as the new output tape, and the current output tape will be rewound to serve as the new input tape.

(a) Assuming that the initial run-construction phase results in 198 runs, how should they be distributed over the three tapes that will become the initial input tapes during the polyphase merge, in order to minimize the number of passes over the data?

(b) How many tape rew windings will occur during the polyphase merge?

84. Do Problems 21.3 and 21.6 from the textbook (page 835).

For November 28, 2018

85. 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)?

86. 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?

87. Suppose you’re planning a trip from Grinnell to Seattle. What practical use might you have for a shortest-path algorithm if you’re planning to drive all the way? What if you plan to travel by air instead?

For November 30, 2018

88. 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?

89. 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 3, 2018

90. Prove that Dijkstra’s algorithm will terminate even if there are some vertices in the graph that are not reachable from the starting vertex by any path at all.

91. 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.

92. Find out who Dijkstra was and how his ideas influenced the practice of computer programming in the twentieth century.
For December 5, 2018

93. 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?

94. Show that the while-loop at lines 15–37 of Figure 14.29 (the Bellman–Ford algorithm) will always either terminate normally or throw an exception, regardless of the structure of the graph and the weights on its edges.

95. For each of the applications described in problems 14.20, 14.24, 14.26, and 14.28 of the textbook (pages 567–569), suggest a way to adapt one of the algorithms implemented in sections 14.2, 14.3, and 14.4 of the textbook to that application.

For December 7, 2018

96. 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?

97. Describe how to adapt the algorithm shown in Figure 14.32 of the textbook to determine the earliest completion time, latest completion time, and slack time for each activity described in an event-node graph.

For December 10, 2018

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

99. Choose one of the three applications described in section 24.2 of the textbook (constructing a maze, finding a minimum spanning tree for a labelled, undirected graph, or finding the nearest common ancestors for two nodes in a tree) and describe (a) the equivalence relation between some elements of the application, (b) the equivalence classes induced by that relation, and (c) how steps in the construction of a solution can be modelled by calls to the union method in an implementation of the DisjointSets class.

100. Describe the operation of path compression in an implementation of the DisjointSets class. In Weiss’s implementation of DisjointSets (Figures 24.20 and 24.21 on pages 911 and 912 of the text), identify the line or lines of Java code that implement path compression.

For December 12, 2018

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

102. Do exercises 1–8 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.

For December 14, 2018

103. List the most important methods and concepts discussed in this course (up to a maximum of five) and briefly summarize the most important thing you have learned about each one.

104. Suggest one or more study questions that, in your judgement, should have appeared on this list but did not.

105. Suggest ways in which this course could be improved the next time I teach it.