1. What are two differences between using `instanceof` and `getClass()` in an `equals()` method?
2. What is the reason that the parameter of an `equals()` method should be of type `Object`?
3. What types of methods can static fields be accessed in?
4. What types of methods can nonstatic fields be accessed in?
5. What is the difference between a checked and an unchecked exception in Java?
6. What is buffering, and why is it used?

7. In what ways can a generic type variable be used inside a class, and how can it not be used?
8. What is the advantage of using generics in Java, as opposed to code that just uses `Object` references to refer to different types of things?
9. How does storing a list in an array compare with using a linked list (i.e., what are the advantages and disadvantages of each)?
10. How do external files compare with storing data in data structures in memory?
11. In the average case, what is the complexity of inserting a value into a binary search tree?
12. In the average case, what is the complexity of searching for an element in a binary search tree?

13. What is the worst–case complexity of searching for an element in a binary search tree?
14. What causes the worst–case behavior of searching for an element in a binary search tree?
15. What is the average case complexity of deleting an element from a binary search tree?
16. Give an operation that is usually efficient for a binary search tree that is not efficient for a heap
17. On average, what is the complexity of inserting an element in a heap?
18. What does Java’s hash code contract say?

19. How could a class violate Java’s hash code contract?

20. What can happen if a class violates Java’s hash code contract?

21. Define a path in a graph.

22. Define a cycle in a graph.

23. Why is graph traversal more difficult than tree traversal?

24. Suppose you have a very large graph, with one vertex for each student in the University, and edges between vertices if two students are siblings. (There are some students at the University who are siblings, but very few compared to the number of students.) Which graph representation(s) would be inefficient in terms of memory use for this situation?

25. How does a singly–linked list compare with a doubly–linked list (Advantages and disadvantages of both?)

26. What is the efficiency of the binary search tree deletion algorithm that we covered?

27. Explain how the heap `getSmallest()` operation works.

28. Give the formula for finding the parent of the element at position *i* of a heap that is stored in an array.

29. What is an upper bound?

30. Why did we not have to do anything to prevent a tree traversal from processing the same elements more than once?

31. Describe what amortized cost is.

32. How does an abstract class compare to an interface?

33. Describe the three levels of copying objects.

34. What has to happens when a hash table that uses open addressing (the open addressing hashing methods we discussed were linear probing or double hashing, as opposed to chained hashing) is full (or close to full), if we want to increase its size so more data can be stored?