1) The ArrayList class implements the \_\_\_\_.

a) Queue interface.

b) Set interface.

c) List interface.

d) Stack interface.

2) A list is a collection that \_\_\_\_.

a) should be used when you need to remember the order of elements in the collection.

b) allows items to be added at one end and removed at the other end.

c) does not allow elements to be inserted in any position.

d) manages associations between keys and values.

3) A stack is a collection that \_\_\_\_.

a) remembers the order of elements, and allows elements to be added and removed only at one end.

b) does not remember the order of elements but allows elements to be added in any position.

c) remembers the order of elements and allows elements to be inserted in any position.

d) remembers the order of elements and allows elements to be inserted only at one end and removed only at the other end.

4) A queue is a collection that \_\_\_\_.

a) remembers the order of elements, and allows elements to be added and removed only at one end.

b) does not remember the order of elements but allows elements to be added in any position.

c) remembers the order of elements and allows elements to be inserted in any position.

d) remembers the order of elements and allows elements to be inserted only at one end and removed only at the other end.

5) A collection without an intrinsic order is called a \_\_\_\_.

a) list

b) stack

c) set

d) queue

6) A collection that allows items to be added only at one end and removed only at the other end is called a \_\_\_\_.

a) list

b) stack

c) set

d) queue

7) A collection that remembers the order of items, and allows items to be added and removed only at one end is called a \_\_\_\_.

a) list

b) stack

c) set

d) queue

8) A collection that allows speedy insertion and removal of already-located elements in the middle of it is called a \_\_\_\_.

a) linked list

b) stack

c) set

d) queue

9) Which data structure would best be used for keeping track of a growing set of groceries to be purchased at the food market?

a) queue

b) stack

c) list

d) array

10) What is included in a linked list node?

I a reference to its neighboring nodes

II an array reference

III a data element

a) I

b) II

c) II and III

d) I and III

11) Which of the following statements about linked lists is correct?

a) Once you have located the correct position, adding elements in the middle of a linked list is inefficient.

b) Visiting the elements of a linked list in random order is efficient.

c) When a node is removed, all nodes after the removed node must be moved down.

d) Linked lists should be used when you know the correct position and need to insert and remove elements efficiently.

12) We might choose to use a linked list over an array list when we will not require frequent \_\_\_\_.

I random access

II inserting new elements

III removing of elements

a) I

b) II

c) III

d) II and III

13) Which nodes need to be updated when we insert a new node to become the fourth node from the beginning of a doubly-linked list?

a) The current third node.

b) The current third and fourth nodes.

c) The current first node.

d) The current fourth and fifth nodes.

14) A binary search requires \_\_\_\_ access.

a) sequential

b) random

c) sorted

d) arbitrary

15) A linear search only requires \_\_\_\_ access.

a) sequential

b) random

c) sorted

d) arbitrary

16) Rather than storing values in an array, a linked list uses a sequence of \_\_\_\_.

a) indexes

b) nodes

c) elements

d) accessors

18) What type of access does a LinkedList provide for its elements?

a) sequential

b) semi-random

c) random

d) sorted

19) Consider the following code snippet:

LinkedList<String> words = new LinkedList<String>();

words.addLast("abc");

words.addLast("def");

words.addLast("ghi");

System.out.print(words.removeLast());

System.out.print(words.removeFirst());

System.out.print(words.removeLast());

What will this code print when it is executed?

a) abcdefghi

b) ghiabcdef

c) abcghidef

d) defghiabc

20) Consider the following code snippet:

LinkedList<String> words = new LinkedList<String>();

words.addFirst("abc");

words.addLast("def");

words.addFirst("ghi");

System.out.print(words.removeLast());

System.out.print(words.removeFirst());

System.out.print(words.removeLast());

What will this code print when it is executed?

a) abcdefghi

b) ghiabcdef

c) abcghidef

d) defghiabc

21) The term \_\_\_\_ is used in computer science to describe an access pattern in which the elements are accessed in arbitrary order.

a) sequential access

b) random access

c) sorted access

d) arbitrary access

22) Which Java package contains the LinkedList class?

a) java.lang

b) java.util

c) java.collections

d) java.io

24) Assume you have created a linked list named myList that currently holds some number of String objects. Which of the following statements correctly adds a new element to the beginning of myList?

a) myList.addFirst("Harry");

b) myList.add("Harry");

c) myList.insert("Harry");

d) myList.put("Harry");

25) Assume you have created a linked list name myList that currently holds some number of String objects. Which of the following statements correctly removes an element from the end of myList?

a) myList.remove();

b) myList.removeLast();

c) myList.getLast();

d) myList.pop();

26) A(n) \_\_\_\_ is a data structure used for collecting a sequence of objects that allows efficient addition and removal of already-located elements in the middle of the sequence.

a) stack

b) queue

c) linked list

d) priority queue

27) What is the meaning of the type parameter E, in the LinkedList<E> code fragment?

a) The elements of the linked list are of class E.

b) The elements of the linked list are of any subclass of class E.

c) The elements of the linked list are any type supplied to the constructor.

d) The elements of the linked list are of class Object.

28) Which method is NOT part of the ListIterator interface?

a) delete

b) add

c) next

d) previous

29) Consider the code snippet shown below. Assume that employeeNames is an instance of type LinkedList<String>.

for (String name : employeeNames)

{

// Do something with name here

}

Which element(s) of employeeNames does this loop process?

a) no elements

b) all elements

c) elements meeting a condition

d) the most recently added elements

30) Which method is NOT part of the ListIterator generic class?

a) hasNext

b) hasMore

c) hasPrevious

d) add

33) Which of the following statements about the LinkedList class is correct?

a) When you use the add method, the new element is inserted before the iterator, and the iterator position is advanced by one position.

b) When you use the add method, the new element is inserted after the iterator, and the iterator position is advanced by one position.

c) When you use the add method, the new element is inserted before the iterator, and the iterator position is not moved

d) When you use the add method, the new element is inserted after the iterator, and the iterator position is not moved.

35) A linked list \_\_\_\_ encapsulates a position anywhere inside the linked list.

a) accessor

b) index

c) operation

d) iterator

36) You use a(n) \_\_\_\_ to access elements inside a linked list.

a) accessor

b) index

c) list iterator

d) queue

37) A linked list allows \_\_\_\_ access, but you need to ask the list for an iterator.

a) sequential

b) random

c) sorted

d) arbitrary

38) The nodes of a(n) \_\_\_\_ linked list class store two links: one to the next element and one to the previous one.

a) array

b) singly

c) doubly

d) randomly

39) Assume you are using a doubly-linked list data structure with many nodes. What is the minimum number of node references that are required to be modified to remove a node from the middle of the list? Consider the neighboring nodes.

a) 1

b) 2

c) 3

d) 4

40) In a linked list data structure, when does the reference to the first node need to be updated?

I inserting into an empty list

II deleting from a list with one node

III deleting an inner node

a) I

b) II

c) I and II

d) III

41) Consider the following code snippet:

LinkedList<String> myLList = new LinkedList<String>();

myLList.add("Mary");

myLList.add("John");

myLList.add("Sue");

ListIterator<String> iterator = myLList.listIterator();

iterator.next();

iterator.next();

iterator.add("Robert");

iterator.previous();

iterator.previous();

iterator.remove();

System.out.println(myLList);

What will be printed when this code is executed?

a) [Mary, John, Robert, Sue]

b) [Mary, John, Sue]

c) [Mary, Robert, Sue]

d) [John, Robert, Sue]

42) Which of the following statements about data structures is correct?

a) Inserting and removing elements that have already been located is faster with a list than with a set.

b) Accessing elements in a linked list in a random fashion is efficient.

c) Adding and removing already-located elements in the middle of a linked list is efficient.

d) A set is an ordered collection of unique elements.

43) Which of the following statements about sets is correct?

a) Inserting and removing elements that have already been located is faster with a list than with a set.

b) A set allows duplicate values.

c) You can add an element to a specific position within a set.

d) A set is a collection of unique elements organized for efficiency.

46) To create a TreeSet for a class of objects, the object class must \_\_\_\_.

a) create an iterator.

b) implement the Comparable interface.

c) implement the Set interface.

d) create a Comparator object.

47) Which of the following statements about manipulating objects in a set is correct?

a) If you try to add an element that already exists, an exception will occur.

b) If you try to remove an element that does not exist, an exception will occur.

c) You can add an element at the position indicated by an iterator.

d) A set iterator visits elements in the order in which the set implementation keeps them.

48) Which of the following statements about manipulating objects in a set is correct?

a) If you try to add an element that already exists, an exception will occur.

b) A set iterator visits elements in the order in which they were added to the set.

c) You can add an element at the position indicated by an iterator.

d) You can remove an element at the position indicated by an iterator.

49) Assume that you have declared a set named mySet to hold String elements. Which of the following statements will correctly insert an element into mySet?

a) mySet.insert("apple");

b) mySet.put(apple");

c) mySet.push("apple");

d) mySet.add("apple");

50) Assume that you have declared a set named mySet to hold String elements. Which of the following statements will correctly remove an element from mySet?

a) mySet.get("apple");

b) mySet.remove("apple");

c) mySet.pop("apple");

d) mySet.delete("apple");

51) Complete the following code snippet, which is intended to determine if a specific value in a variable named targetWord appears in a set of String values named mySet:

for (String aWord : mySet)

{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

{

System.out.println ("The word " + targetWord + " was found.");

}

)

a) if (mySet.equalsIgnoreCase(targetWord))

b) if (mySet == targetWord)

c) if (mySet.contains(targetWord))

d) if (mySet.get(targetWord))

52) Which of the following statements about manipulating objects in a map is NOT correct?

a) Use the add method to add a new element to the map.

b) Use the get method to retrieve a value from the map.

c) Use the keyset method to get the set of keys for the map.

d) Use the remove method to remove a value from the map.

53) Complete the following code, which is intended to print out all key/value pairs in a map named myMap that contains String data for student IDs and names:

Map<String, String> myMap = new HashMap<String, String>();

. . .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

for (String aKey : mapKeySet)

{

String name = myMap.get(aKey);

System.out.println("ID: " + aKey + "->" + name);

}

a) Map<String, String> mapKeySet = myMap.keySet();

b) Set<String, String> mapKeySet = myMap.keySet();

c) Set<String> mapKeySet = myMap.getKeySet();

d) Set<String> mapKeySet = myMap.keySet();

54) Complete the following code, which is intended to print out all key/value pairs in a map named myMap that contains String data for student IDs and names:

Map<String, String> myMap = new HashMap<String, String>();

. . .

Set<String> mapKeySet = myMap.keySet();

for (String aKey : mapKeySet)

{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

System.out.println("ID: " + aKey + "->" + name);

}

a) String name = myMap.get(aKey);

b) String name = myMap.next(aKey);

c) String name = MapKeySet.get(aKey);

d) String name = MapKeySet.next(aKey);

55) Assume that you have declared a map named myMap to hold String elements with Integer keys. Which of the following statements will correctly insert an element into myMap?

a) myMap.insert(3, "apple");

b) myMap.put(3, "apple");

c) myMap.push(3, "apple");

d) myMap.add(3, "apple");

56) Assume that you have declared a map named myMap to hold String elements with Integer keys. Which of the following statements will correctly remove an element from myMap?

a) myMap.get(3);

b) myMap.remove(3);

c) myMap.pop(3);

d) myMap.delete(3);

57) Assume that you have declared a map named myMap to hold String elements with Integer keys. Which of the following statements will correctly retrieve the value of an element from myMap by using its key?

a) myMap.get("apple");

b) myMap.peek("apple");

c) myMap.get(3);

d) myMap.peek(3);

58) Which of the following statements about manipulating objects in a map is NOT correct?

a) If you attempt to retrieve a value with a key that is not associated with any value, you will receive a null result.

b) You cannot change the value of an existing association in the map; you must delete it and re-add it with the new values.

c) Use the get method to retrieve a value associated with a key in the map.

d) Use the put method to add an element to the map.

59) Consider the following code snippet:

Map<String, Integer> scores;

If you need to visit the keys in sorted order, which of the following statements will create a structure to support this?

a) scores = new HashMap<String, Integer>;

b) scores = new TreeMap<String, Integer>;

c) scores = new Map<String, Integer>;

d) scores = new HashTable<String, Integer>;

60) Consider the following code snippet:

Map<String, Integer> scores;

You expect to retrieve elements randomly by key, and want fastest retrieval times. Which of the following statements will create a structure to support this?

a) scores = new HashMap<String, Integer>;

b) scores = new TreeMap<String, Integer>;

c) scores = new Map<String, Integer>;

d) scores = new TreeSet<String, Integer>;

61) You want to enumerate all of the keys in a map named myMap whose keys are type String. Which of the following statements will allow you to do this?

a)

Set<String> keySet = myMap.keySet();

for (String key : keySet) {. . . }

b)

Set<String> keySet = myMap.getKeys();

for (String key : keySet) {. . . }

c)

Set<String> keySet = myMap.keys();

for (String key : keySet) {. . . }

d)

Set<String> keySet = myMap.getKeySet();

for (String key : keySet) {. . . }

62) You need to access values by an integer position. Which collection type should you use?

a) Map

b) Hashtable

c) ArrayList

d) Queue

63) You need to access values in objects by a key that is not part of the object. Which collection type should you use?

a) Map

b) Hashtable

c) ArrayList

d) Queue

64) You need to access values in the order in which they were added (first in, first out), and not randomly. Which collection type should you use?

a) Map

b) Hashtable

c) Stack

d) Queue

65) You need to access values in the opposite order in which they were added (last in, first out), and not randomly. Which collection type should you use?

a) Map

b) Hashtable

c) Stack

d) Queue

66) You need to access values using a key, and the keys must be sorted. Which collection type should you use?

a) TreeMap

b) ArrayList

c) HashMap

d) Queue

67) You need to access values by their position. Which collection type should you use?

a) TreeSet

b) ArrayList

c) Stack

d) Queue

68) You have decided to store objects of a class in a TreeSet structure. Which of the following statements is correct?

a) If the object class implements the Comparable interface, and the sort order in the compare method is acceptable, you do not have to do anything else.

b) If the object class implements the Comparable interface, and the sort order in the compareTo method is acceptable, you do not have to do anything else.

c) If the object class implements the Comparable interface, and the sort order in the compare method is acceptable, you must create a comparator object.

d) If the object class implements the Comparable interface, and the sort order in the compareTo method is acceptable, you must create a comparator object.

69) Which data structure would best be used for storing a set of numbers and sorting them in ascending order?

a) queue

b) stack

c) list

d) array