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) 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

4) 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.

5) 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

6) 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.

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

a) sequential

b) random

c) sorted

d) arbitrary

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

a) sequential

b) random

c) sorted

d) arbitrary

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

a) indexes

b) nodes

c) elements

d) accessors

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

a) sequential

b) semi-random

c) random

d) sorted

11) 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

12) 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

13) 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

14) Which Java package contains the LinkedList class?

a) java.lang

b) java.util

c) java.collections

d) java.io

15) 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");

16) 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();

17) 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.

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

a) delete

b) add

c) next

d) previous

19) 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

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

a) hasNext

b) hasMore

c) hasPrevious

d) add

21) 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.

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

a) accessor

b) index

c) operation

d) iterator

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

a) sequential

b) random

c) sorted

d) arbitrary

24) 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

25) 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

26) 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

27) 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]