Course Title: Big Java, Early Objects

Chapter Number: 15 The Java Collections Framework

Question type: Multiple Choice

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

a) Queue interface.

b) Set interface.

c) List interface.

d) Stack interface.

Answer: c

Title: The ArrayList class \_\_\_\_.

Difficulty: Easy

Section Reference 1: 15.1 An Overview of the Collections Framework

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.

Answer: a

Title: A list is a collection that \_\_\_\_.

Difficulty: Easy

Section Reference 1: 15.1 An Overview of the Collections Framework

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.

Answer: a

Title: A stack is a collection that \_\_\_\_.

Difficulty: Easy

Section Reference 1: 15.1 An Overview of the Collections Framework

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.

Answer: d

Title: A queue is a collection that \_\_\_\_.

Difficulty: Easy

Section Reference 1: 15.1 An Overview of the Collections Framework

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

a) list

b) stack

c) set

d) queue

Answer: c

Title: A collection without an intrinsic order is called a \_\_\_\_.

Difficulty: Easy

Section Reference 1: 15.1 An Overview of the Collections Framework

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

Answer: d

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

Difficulty: Easy

Section Reference 1: 15.1 An Overview of the Collections Framework

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

Answer: b

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

Difficulty: Easy

Section Reference 1: 15.1 An Overview of the Collections Framework

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

Answer: a

Title: A collection that allows speedy insertion and removal of elements in the middle of it is called a \_\_\_\_.

Difficulty: Easy

Section Reference 1: 15.1 An Overview of the Collections Framework

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

Answer: c

Title: Which data structure would best be used for keeping track of groceries to be purchased?

Difficulty: Medium

Section Reference 1: 15.1 An Overview of the Collections Framework

10) Select an appropriate expression to complete the following code segment, which is designed to print a message if the string stored in name is part of the players collection.

Collection<String> players = new ArrayList<String>();

// code to add elements to the collection here

if \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

System.out.print(name + " is one of the players in the collection.");

a) (players.indexOf(name))

b) (players.contains(name))

c) (players.search(name))

d) (players.equals(name))

Answer: b

Title: Select expression to determine if a string is part of a collection

Difficulty: Easy

Section Reference 1: 15.1 An Overview of the Collections Framework

11) 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

Answer: d

Title: What is included in a linked list node?

Difficulty: Easy

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.1 The Structure of Linked Lists

12) 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.

Answer: d

Title: Which statement about linked lists is correct?

Difficulty: Easy

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.1 The Structure of Linked Lists

13) 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

Answer: a

Title: We might choose to use a linked list over an array list because we will not require frequent \_\_\_\_.

Difficulty: Easy

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.1 The Structure of Linked Lists

14) 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.

Answer: b

Title: Which nodes need to be updated when we insert a new node as the fourth node from the beginning of a single-linked list?

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.1 The Structure of Linked Lists

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

a) sequential

b) random

c) sorted

d) arbitrary

Answer: b

Title: A binary search requires \_\_\_\_\_\_\_\_ access.

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.1 The Structure of Linked Lists

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

a) sequential

b) random

c) sorted

d) arbitrary

Answer: a

Title: A linear search requires \_\_\_\_ access.

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.1 The Structure of Linked Lists

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

a) indexes

b) nodes

c) elements

d) accessors

Answer: b

Title: A linked list uses a sequence of \_\_\_\_\_\_.

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.1 The Structure of Linked Lists

18) Which of the following algorithms would be efficiently executed using a LinkedList?

a) Tracking paths in a maze.

b) Binary search.

c) Remove first *n*/ 2 elements from a list of *n* elements.

d) Read *n* / 2 elements in random order from a list of *n* elements.

Answer: c

Title: Which algorithm would be efficiently executed using a LinkedList?

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.1 The Structure of Linked Lists

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

a) sequential

b) semi-random

c) random

d) sorted

Answer: a

Title: What type of access does a LinkedList provide for its elements?

Difficulty: Easy

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.1 The Structure of Linked Lists

20) 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

Answer: b

Title: What does the following LinkedList code print?

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.1 The Structure of Linked Lists

21) 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

Answer: d

Title: What does the following LinkedList code print?

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.1 The Structure of Linked Lists

22) 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

Answer: b

Title: The term \_\_\_ describes a pattern of accessing elements in an arbitrary order.

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.1 The Structure of Linked Lists

23) Which Java package contains the LinkedList class?

a) java.lang

b) java.util

c) java.collections

d) java.io

Answer: b

Title: Which Java package contains the LinkedList class?

Difficulty: Easy

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.2 The LinkedList Class of the Java Collections Framework

24) What can a generic class be parameterized for?

a) properties

b) iterators

c) type

d) methods

Answer: c

Title: What can a generic class can be parameterized for?

Difficulty: Easy

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.2 The LinkedList Class of the Java Collections Framework

25) 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");

Answer: a

Title: Which statement correctly adds a new element to the beginning of a linked list?

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.2 The LinkedList Class of the Java Collections Framework

26) 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();

Answer: b

Title: Which statement correctly removes an element from the end of a linked list?

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2:15.2.2 The LinkedList Class of the Java Collections Framework

27) 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

Answer: c

Title: A \_\_\_\_ allows for ... efficient addition and removal ... in the middle of the sequence.

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.2 The LinkedList Class of the Java Collections Framework

28) 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.

Answer: c

Title: What is the meaning of E, in the LinkedList<E> code fragment?

Difficulty: Easy

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.2 The LinkedList Class of the Java Collections Framework

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

a) delete

b) add

c) next

d) previous

Answer: a

Title: Which method is NOT part of the ListIterator interface?

Difficulty: Easy

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.3 List Iterators

30) 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

Answer: b

Title: Which elements of employeeNames does this loop process?

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.3 List Iterators

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

a) hasNext

b) hasMore

c) hasPrevious

d) add

Answer: b

Title: Which method is NOT part of the ListIterator generic class?

Difficulty: Easy

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.3 List Iterators

32) Select an appropriate expression to complete the following code segment, which is designed to print a message if the string stored in name is the first element of the players linked list.

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

// code to add elements to the linked list

if \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

System.out.print(name + " is the first player on the list.");

a) (players.indexOf(name) == 1)

b) (players.contains(name))

c) (players.getFirst().equals(name))

d) (players[0].equals(name))

Answer: c

Title: Select expression to determine if a string is the first element of a linked list

Difficulty: Easy

Section Reference 1: 15.2 Linked Lists

33) Select an appropriate expression to complete the method below. The method should return the number of times that the string stored in name appears in theList.

public static int count(LinkedList<String> theList, String name)

{

int number = 0;

Iterator<String> iter = theList.iterator();

while (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

{

if (iter.next().equals(name))

{

number++;

}

}

return number;

}

a) iter.hasNext()

b) iter.next() != null

c) theList.hasNext()

d) theList.next() != null

Answer: a

Title: Select expression to complete method that counts number of times a string appears in linked list

Difficulty: Easy

Section Reference 1: 15.2 Linked Lists

34) Select an appropriate expression to complete the following method, which is designed to visit the elements in theList and replace each occurrence of the string "hello" with the string "goodbye".

public static void helloGoodbye(LinkedList<String> theList)

{

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

while (iterator.hasNext())

{

if (iterator.next().equals("hello"))

{

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

}

}

}

a) iterator.replace("hello", "goodbye");

b) iterator.next() = "goodbye";

c) iterator.previous("goodbye");

d) iterator.set("goodbye");

Answer: d

Title: Select statement to complete method that replaces every occurrence of a string in a list

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

35) When using a list iterator, on which condition will the IllegalStateException be thrown?

a) Calling remove after calling next.

b) Calling next after calling previous.

c) Calling remove after calling remove.

d) Calling remove after calling previous.

Answer: c

Title: On which condition will the IllegalStateException be thrown?

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.3 List Iterators

36) When using a list iterator, on which condition will the IllegalStateException be thrown?

a) Calling remove after calling next.

b) Calling add after calling previous.

c) Calling remove after calling add.

d) Calling previous after calling previous.

Answer: c

Title: On which condition will the IllegalStateException be thrown?

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.3 List Iterators

37) 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.

Answer: a

Title: Which statement about the LinkedList class is correct?

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.3 List Iterators

38) When using a list iterator, on which condition will the NoSuchElementException be thrown?

a) Calling next when you are past the end of the list.

b) Calling next when the iterator points to the last element.

c) Calling remove after calling add.

d) Calling remove after calling previous.

Answer: a

Title: On which condition will the NoSuchElementException be thrown?

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.3 List Iterators

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

a) accessor

b) index

c) operation

d) iterator

Answer: d

Title: A linked list \_\_\_\_ encapsulates a position inside the linked list.

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.3 List Iterators

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

a) accessor

b) index

c) list iterator

d) queue

Answer: c

Title: You use a \_\_\_\_ to access elements inside a linked list.

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.3 List Iterators

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

a) sequential

b) random

c) sorted

d) arbitrary

Answer: a

Title: A linked list allows \_\_\_\_\_\_ access, but you need to ask for an iterator.

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.3 List Iterators

42) 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

Answer: c

Title: The nodes of a \_\_\_ linked list class store two links…

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.3 List Iterators

43) 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

Answer: b

Title: How many references need to be updated when you remove from the middle of a linked list?

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.3 List Iterators

44) 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

Answer: c

Title: In a linked list, when does the reference to the first node need to be updated?

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2: 15.2.3 List Iterators

45) 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]

Answer: c

Title: What will be printed when this code is executed?

Difficulty: Medium

Section Reference 1: 15.2 Linked Lists

Section Reference 2:15.2.3 List Iterators

46) 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.

Answer: c

Title: Which statement about data structures is correct?

Difficulty: Medium

Section Reference 1: 15.3 Sets

Section Reference 2: 15.3.1 Choosing a Set Implementation

47) 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.

Answer: d

Title: Which statement about sets is correct?

Difficulty: Medium

Section Reference 1: 15.3 Sets

Section Reference 2: 15.3.1 Choosing a Set Implementation

48) Which of the following statements about hash tables is NOT correct?

a) Elements are grouped into smaller collections that share the same characteristic.

b) You can form hash tables holding objects of type String.

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

d) The value used to locate an element in a hash table is called a hash code.

Answer: c

Title: Which statement about hash tables is NOT correct?

Difficulty: Medium

Section Reference 1: 15.3 Sets

Section Reference 2: 15.3.1 Choosing a Set Implementation

49) Which of the following statements about the TreeSet class is NOT correct?

a) Elements are stored in sorted order.

b) Elements are arranged in linear fashion.

c) Elements are stored in nodes.

d) To use a TreeSet, it must be possible to compare the elements.

Answer: b

Title: Which statement about the TreeSet class is NOT correct?

Difficulty: Medium

Section Reference 1: 15.3 Sets

Section Reference 2: 15.3.1 Choosing a Set Implementation

50) Select an appropriate declaration to complete the following code segment, which is designed to read strings from standard input and display them in increasing alphabetical order, excluding any duplicates.

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

Scanner input = new Scanner(System.in);

while (input.hasNext())

{

words.add(input.next());

}

System.out.print(words);

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

b) Set<String> words = new Set<String>();

c) Set<String> words = new HashSet<String>();

d) Set<String> words = new TreeSet<String>();

Answer: d

Title: Select appropriate declaration to complete code segment

Difficulty: Medium

Section Reference 1: 15.3 Sets

51) Select an appropriate expression to complete the following method, which is designed to return the number of elements in the parameter array numbers. If a value appears more than once, it should be counted exactly once.

public static int countElementsOnce(int[] numbers)

{

Set<Integer> values = new HashSet<Integer>();

for (int num: numbers)

{

values.add(num);

}

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

}

a) return numbers.length;

b) return values.length();

c) return values.size();

d) return numbers.length – values.size();

Answer: c

Title: Select statement to complete method that counts elements once in an integer array

Difficulty: Hard

Section Reference 1: 15.3 Sets

52) 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.

Answer: b

Title: To create a TreeSet for a class of objects, the class must \_\_\_\_.

Difficulty: Medium

Section Reference 1: 15.3 Sets

Section Reference 2: 15.3.1 Choosing a Set Implementation

53) 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.

Answer: d

Title: Which statement about manipulating objects in a set is correct?

Difficulty: Medium

Section Reference 1: 15.3 Sets

Section Reference 2: 15.3.2 Working with Sets

54) 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.

Answer: d

Title: Which statement about manipulating objects in a set is correct?

Difficulty: Medium

Section Reference 1: 15.3 Sets

Section Reference 2: 15.3.2 Working with Sets

55) 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");

Answer: d

Title: How to insert an element into a set

Difficulty: Easy

Section Reference 1: 15.3 Sets

Section Reference 2: 15.3.2 Working with Sets

56) 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");

Answer: b

Title: How to remove an element from a set

Difficulty: Easy

Section Reference 1: 15.3 Sets

Section Reference 2: 15.3.2 Working with Sets

57) 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))

Answer: c

Title: How to determine if a value is in a set

Difficulty: Medium

Section Reference 1: 15.3 Sets

Section Reference 2: 15.3.2 Working with Sets

58) 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.

Answer: a

Title: Which statement about manipulating objects in a map is NOT correct?

Difficulty: Medium

Section Reference 1: 15.4 Maps

59) 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();

Answer: d

Title: Complete the code to enumerate values in a map

Difficulty: Medium

Section Reference 1: 15.4 Maps

60) 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);

Answer: a

Title: Complete the code to enumerate values in a map

Difficulty: Medium

Section Reference 1: 15.4 Maps

61) 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");

Answer: b

Title: How to insert an element into a map

Difficulty: Easy

Section Reference 1: 15.4 Maps

62) 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);

Answer: b

Title: How to remove an element from a map

Difficulty: Easy

Section Reference 1: 15.4 Maps

63) 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);

Answer: c

Title: How to retrieve an element from a map

Difficulty: Easy

Section Reference 1: 15.4 Maps

64) Your program uses a Map structure to store a number of user ids and corresponding email addresses. Although each user must have a unique id, two or more users can share the same email address. Select an appropriate expression to complete the method below, which adds a new id and email address to the map only if the id is not already in use. If the id is already in use, an error message is printed.

public static void addUserID(Map<String, String> users, String id, String email)

{

String currentEmail = users.get(id);

if (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

{

users.put(id, email);

}

else

{

System.out.println(id + " is already in use.");

}

}

a) currentEmail.equals(email)

b) !currentEmail.equals(email)

c) currentEmail == null

d) currentEmail != null

Answer: c

Title: Select expression to complete method that adds unique ids to a map

Difficulty: Hard

Section Reference 1: 15.4 Maps

65) 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.

Answer: b

Title: Which statement about manipulating objects in a map is NOT correct?

Difficulty: Medium

Section Reference 1: 15.4 Maps

66) 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>;

Answer: b

TitleWhich statement creates a structure to support visiting elements in sorted order?

Difficulty: Medium

Section Reference 1: 15.4 Maps

67) 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>;

Answer: a

Title: Which statement creates a structure to support visiting elements in random order?

Difficulty: Medium

Section Reference 1: 15.4 Maps

68) 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) {. . . }

Answer: a

Title: Which statement enumerates keys in a map structure?

Difficulty: Medium

Section Reference 1: 15.4 Maps

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

a) Map

b) Hashtable

c) ArrayList

d) Queue

Answer: c

Title: Which collection type should you use to access values by position?

Difficulty: Medium

Section Reference 1: 15.4 Maps

Section Reference 2: How To 15.1 Choosing a Collection

70) 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

Answer: a

Title: Which collection type should you use?

Difficulty: Medium

Section Reference 1: 15.4 Maps

Section Reference 2: How To 15.1 Choosing a Collection

71) 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

Answer: d

Title: Which collection type should you use?

Difficulty: Medium

Section Reference 1: 15.4 Maps

Section Reference 2: How To 15.1 Choosing a Collection

72) 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

Answer: c

Title: Which collection type should you use?

Difficulty: Medium

Section Reference 1: 15.4 Maps

Section Reference 2: How To 15.1 Choosing a Collection

73) 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

Answer: a

Title: Which collection type should you use?

Difficulty: Medium

Section Reference 1: 15.4 Maps

Section Reference 2: How To 15.1 Choosing a Collection

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

a) TreeSet

b) ArrayList

c) Stack

d) Queue

Answer: b

Title: Which collection type should you use?

Difficulty: Medium

Section Reference 1: 15.4 Maps

Section Reference 2: How To 15.1 Choosing a Collection

74) 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.

Answer: b

Title: Which statement about tree sets is correct?

Difficulty: Medium

Section Reference 1: 15.4 Maps

Section Reference 2: How To 15.1 Choosing a Collection

75) 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

Answer: d

Title: Which data structure would be best for storing a set of numbers and sorting them in ascending order?

Difficulty: Medium

Section Reference 1: 15.4 Maps

Section Reference 2: How To 15.1 Choosing a Collection

76) Which of the following algorithms would be efficiently executed on an ArrayList?

a) add 1 element to the middle of a list with *n* elements

b) add *n* / 2 elements to a list with *n* / 2 elements

c) remove first *n* / 2 elements from a list of *n* elements

d) read *n* / 2 elements in random order from a list of *n* elements

Answer: d

Title: Which algorithm would be efficiently executed on an ArrayList?

Difficulty: Hard

Section Reference 1: 15.4 Maps

Section Reference 2: How To 15.1 Choosing a Collection

77) What operation is least efficient in a LinkedList?

a) Adding an element in a position that has already been located.

b) Linear traversal step.

c) Removing an element when the element's position has already been located.

d) Random access of an element.

Answer: d

Title: What operation is least efficient in a LinkedList?

Difficulty: Medium

Section Reference 1: 15.4 Maps

Section Reference 2: How To 15.1 Choosing a Collection

78) You need to write a program to simulate the effect of adding an additional cashier in a supermarket to reduce the length of time customers must wait to check out. Which data structure would be most appropriate to simulate the waiting customers?

a) map

b) stack

c) queue

d) linked list

Answer: c

Title: Which data structure should be used to model this situation?

Difficulty: Medium

Section Reference 1: 15.4 Maps

Section Reference 2: How To 15.1 Choosing a Collection

79) You need to write a program to build and maintain an address book. Since the program must support the possibility of having duplicate names, which data structure would be most appropriate to model this situation?

a) map

b) stack

c) queue

d) linked list

Answer: d

Title: Which data structure should be used to model this situation?

Difficulty: Medium

Section Reference 1: 15.4 Maps

Section Reference 2: How To 15.1 Choosing a Collection

80) You need to write a program to build and maintain a catalog of college courses that are associated with specific majors. Which data structure would be most appropriate to model this situation?

a) map

b) stack

c) queue

d) linked list

Answer: a

Title: Which data structure should be used to model this situation?

Difficulty: Medium

Section Reference 1: 15.4 Maps

Section Reference 2: How To 15.1 Choosing a Collection

81) You need to write a program to manage a waiting list of patrons at a restaurant. Which data structure would be most appropriate to model this situation?

a) map

b) stack

c) queue

d) linked list

Answer: c

Title: Which data structure should be used to model this situation?

Difficulty: Medium

Section Reference 1: 15.4 Maps

Section Reference 2: How To 15.1 Choosing a Collection

82) You intend to use a hash set with your own object class. Which of the following statements is NOT correct?

a) You do not have to do anything additional. You can use the hashCode function of the Object class.

b) You can create your own function to compute a hashCode value.

c) You can override the hashCode method in the Object class to provide your own hashCode method.

d) Your class's hashCode method does not need to be compatible with its equals method.

Answer: d

Title: Which statement about hash sets is NOT correct?

Difficulty: Medium

Section Reference 1: 15.4 Maps

Section Reference 2: Special Topic 15.1 Hash Functions

83) Which of the following statements about hash functions is NOT correct?

a) A hash function produces a unique integer-valued hash code value for each distinct object.

b) A good hash function will minimize the number of objects that are assigned the same hash code.

c) Using a prime number as a hash multiplier will produce better hash codes.

d) If you supply your own hashCode method for a class, it must be compatible with that class's equals method.

Answer: a

Title: Which statement about hash functions is NOT correct?

Difficulty: Medium

Section Reference 1: 15.4 Maps

Section Reference 2: Special Topic 15.1 Hash Functions

84) Which of the following statements about stacks is correct?

a) A stack implements first-in, first-out retrieval.

b) A stack implements random retrieval.

c) A stack implements last-in, first-out retrieval.

d) A stack stores elements in sorted order.

Answer: c

Title: Which statement about stacks is correct?

Difficulty: Medium

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.1 Stacks

85) An Undo feature in a word processor program that allows you to reverse a previously completed command is probably implemented using which structure type?

a) queue

b) linked list

c) stack

d) hash table

Answer: c

Title: An Undo feature is probably implemented using which structure type?

Difficulty: Medium

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.1 Stacks

86) Which of the following correctly declares a stack that will hold String elements?

a) Stack<String> s = new Stack<String>();

b) Stack s = new Stack<String>();

c) String s = new Stack<String>();

d) String s = new Stack();

Answer: a

Title: How to declare a stack that will hold String elements

Difficulty: Medium

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.1 Stacks

87) Assume that you have declared a stack named myStack to hold String elements. Which of the following statements will correctly add an element to myStack?

a) myStack.put("apple");

b) myStack.addItem("apple");

c) myStack.push("apple");

d) myStack.insert("apple");

Answer: c

Title: How to add an element to a stack

Difficulty: Easy

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.1 Stacks

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

a) myStack.remove();

b) myStack.get();

c) myStack.delete();

d) myStack.pop();

Answer: d

Title: How to remove an element from a stack

Difficulty: Easy

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.1 Stacks

89) Assume that you have declared a stack named myStack to hold String elements. Which of the following statements will correctly retrieve the top element from myStack without removing it?

a) myStack.peek();

b) myStack.get();

c) myStack.next();

d) myStack.pop();

Answer: a

Title: How to retrieve an element from a stack without removing it

Difficulty: Easy

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.1 Stacks

90) Consider the code snippet shown below:

Stack<String> words1 = new Stack<String>();

Stack<String> words2 = new Stack<String>();

words1.push("abc");

words1.push("def");

words1.push("ghi");

while (!words1.empty())

{

words2.push(words1.pop());

}

while (!words2.empty())

{

System.out.print(words2.pop());

}

What will be printed when this code is executed?

a) abcdefghi

b) ghiabcdef

c) abcghidef

d) defghiabc

Answer: a

Title: What does the following Stack code print?

Difficulty: Medium

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.1 Stacks

91) Consider the following code snippet:

Stack<String> words1 = new Stack<String>();

ArrayList<String> words3 = new ArrayList<String>();

words1.push("abc");

words1.push("def");

words1.push("ghi");

while (!words1.empty())

{

words3.add(words1.pop());

}

int i = words3.size() - 1;

while (i >= 0)

{

System.out.print(words3.remove(i));

i--;

}

What will this code print when it is executed?

a) abcdefghi

b) ghiabcdef

c) abcghidef

d) defghiabc

Answer: a

Title: What does the following Stack and ArrayList code print?

Difficulty: Medium

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.1 Stacks

92) Which data structure would best be used for finding a path out of a maze?

a) queue

b) stack

c) list

d) array

Answer: b

Title: Which data structure would best be used for finding a path out of a maze?

Difficulty: Medium

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.1 Stacks

93) Which operations from the list data structure could be used to implement the push and pop operations of a stack data structure?

I addLast

II addFirst

III removeFirst

a) I

b) II

c) I and II

d) II and III

Answer: d

Title: Which list operations could be used to implement push and pop operations of a stack?

Difficulty: Medium

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.1 Stacks

94) Consider the following code snippet:

Stack<String> stringStack = new Stack<String>();

stringStack.push("ab");

stringStack.push("abc");

stringStack.push("a");

while (stringStack.size() > 0)

{

System.out.print(stringStack.pop() + ",");

}

What output will be produced when this code is executed?

a) ab,abc,a,

b) a,abc,ab,

c) a,ab,abc,

d) abc,ab,a,

Answer: b

Title: Which output will be produced from this stack code?

Difficulty: Medium

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.1 Stacks

95) Print jobs submitted to a printer would probably be stored in which type of data structure?

a) queue

b) linked list

c) stack

d) hash table

Answer: a

Title: Print jobs are probably stored using which structure type?

Difficulty: Easy

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.2 Queues

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

a) myQueue.remove();

b) myQueue.get();

c) myQueue.delete();

d) myQueue.pop();

Answer: a

Title: How to remove an element from a queue

Difficulty: Easy

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.2 Queues

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

a) myQueue.insert("apple");

b) myQueue.put("apple");

c) myQueue.push("apple");

d) myQueue.add("apple");

Answer: d

Title: How to insert an element into a queue

Difficulty: Easy

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.2 Queues

98) Select an appropriate expression to complete the method below, which is designed to print the element at the bottom of a Stack collection. The contents of the original stack are restored before the method terminates. It is safe to assume that the original stack contains at least one element.

public static void printBottom(Stack<String> theStack)

{

Stack<String> anotherStack = new Stack<String>();

while (theStack.size() > 0)

{

anotherStack.push(theStack.pop());

}

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

while (anotherStack.size() > 0)

{

theStack.push(anotherStack.pop());

}

}

a) System.out.println(theStack.pop());

b) System.out.println(theStack.peek());

c) System.out.println(anotherStack.peek());

d) System.out.println(anotherStack.pop());

Answer: c

Title: Select expression to complete method that prints element at the bottom of a stack

Difficulty: Medium

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

99) Assuming that names is a Queue of String objects, select a statement to complete the code segment below. The code is designed to remove the last element from the queue, which is guaranteed to have at least one element.

Queue<String> aQueue = new LinkedList<String>();

while (names.size() > 1)

{

aQueue.add(names.remove());

}

names.remove();

while (aQueue.size() > 0)

{

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

}

a) names.add(aQueue.remove());

b) names.add(aQueue.peek());

c) aQueue.add(names.remove());

d) aQueue.add(names.peek());

Answer: a

Title: Complete code segment to remove the last element in a Queue object.

Difficulty: Medium

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

100) Select an appropriate expression to complete the following method, which is designed to return the sum of the two smallest values in the parameter array numbers.

public static int sumTwoLowestElements(int[] numbers)

{

PriorityQueue<Integer> values = new PriorityQueue<Integer>();

for (int num: numbers)

{

values.add(num);

}

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

}

a) return values.peek() + values.peek();

b) return values.peek() + values.remove();

c) return values.remove() + values.remove();

d) return values.remove() \* 2;

Answer: c

Title: Select statement to complete method that returns sum of two lowest elements in an array

Difficulty: Hard

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

101) Suppose you push integer elements 1,2,3,4 onto a stack in that order. Then pop an element off the stack and add that element to a queue. You repeat that process three more times. In what order will you remove the elements from the queue?

a) 1,2,3,4

b) 1,2,4,3

c) 4,3,2,1

d) 4,3,1,2

Answer: c

Title: In what order will you remove the elements from this queue?

Difficulty: Medium

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.2 Queues

102) Which data structure would best be used for keeping track of bank customers waiting for a teller?

a) queue

b) stack

c) list

d) array

Answer: a

Title: Which data structure would best be used for keeping track of bank customers waiting for a teller?

Difficulty: Easy

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.2 Queues

103) Suppose we have two String objects and treat the characters in each string from beginning to end in the following way: With one string, we push each character on a stack. With the other string, we add each character to a queue. After processing both strings, we then pop one character from the stack and remove one character from the queue, and compare the pair of characters to each other. We do this until the stack and the queue are both empty. What does it mean if all the character pairs match?

a) The strings are the identical.

b) The strings are different.

c) One string is the reverse of the other.

d) We can only conclude the strings are of the same length

Answer: c

Title: What can we conclude after inserting and removing characters in this fashion?

Difficulty: Medium

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.2 Queues

104) Consider the following code snippet:

Queue<String> stringQueue = new LinkedList<String>();

stringQueue.add("ab");

stringQueue.add("abc");

stringQueue.add("a");

while (stringQueue.size() > 0)

{

System.out.print(stringQueue.remove() + ",");

}

What output will be produced when this code is executed?

a) ab,abc,a,

b) a,abc,ab,

c) a,ab,abc,

d) abc,ab,a,

Answer: a

Title: Which output will be produced from this queue code?

Difficulty: Medium

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.2 Queues

105) Suppose we create a deque (double-ended queue) data structure. It is basically a queue, with its addLast and removeFirst operations, but we also add the addFirst and removeLast operations. Which of the following is best modeled by the deque data structure?

a) A toll booth on a highway.

b) A cross country race.

c) A computer keyboard typing buffer.

d) A Memorial Day parade.

Answer: c

Title: Which of the following is best modeled by the deque data structure?

Difficulty: Hard

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.2 Queues

106) Suppose we create a deque (double-ended queue) data structure. It is basically a queue, with its addLast and removeFirst operations, but we also add the addFirst and removeLast operations. If a line at the grocery store resembles a deque more than a queue, what is the most likely reason?

I the cashier is very fast

II the line is very long

III the cashier is very slow

a) I

b) II

c) I and II

d) II and III

Answer: d

Title: If a line at the grocery store resembles a deque more than a queue, what is the most likely reason?

Difficulty: Hard

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.2 Queues

107) Which of the following statements about a priority queue structure is correct?

a) It uses a FIFO discipline.

b) New items must be inserted at the end of the queue.

c) Elements must be removed in priority order.

d) It uses a LIFO discipline.

Answer: c

Title: Which statement about a priority queue structure is correct?

Difficulty: Medium

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.3 Priority Queues

108) Which of the following statements about a priority queue structure is NOT correct?

a) Elements added to a priority queue must belong to a class that implements the Comparable interface.

b) New elements can be inserted in any order.

c) The remove method is used to remove an element from the priority queue.

d) The insert method is used to add a new element to the priority queue.

Answer: d

Title: Which statement about a priority queue structure is NOT correct?

Difficulty: Medium

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.3 Priority Queues

109) Consider the following code snippet:

PriorityQueue<String> stringQueue = new PriorityQueue<String>();

stringQueue.add("ab");

stringQueue.add("abc");

stringQueue.add("a");

while (stringQueue.size() > 0)

{

System.out.print(stringQueue.remove() + ",");

}

What output will be produced when this code is executed?

a) ab,abc,a,

b) a,abc,ab,

c) a,ab,abc,

d) abc,ab,a,

Answer: c

Title: Which output will be produced from this priority queue code?

Difficulty: Medium

Section Reference 1: 15.5 Stacks, Queues, and Priority Queues

Section Reference 2: 15.5.3 Priority Queues