**Question 1**

|  |  |  |
| --- | --- | --- |
|  | • | Entries are not organized as key/value pairs. |
|  | • | Duplicate entries are rejected. |
|  | • | Entries are sorted using a Comparator or the Comparable interface. |

Which interface of the java.util package offers the specified behavior?

|  |  |
| --- | --- |
| a. | List |
| b. | Map |
| c. | Set |
| d. | SortedSet |
| e. | SortedMap |
| f. | None of the above |

**Question 2**

import java.util.\*;

class GFC111 {

public static void main (String[] args) {

Object m = new LinkedHashMap();

System.out.print((m instanceof Collection)+",");

System.out.print((m instanceof Map)+",");

System.out.print(m instanceof HashMap);

}}

What is the result of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | Prints: false,false,false |
| b. | Prints: false,false,true |
| c. | Prints: false,true,false |
| d. | Prints: false,true,true |
| e. | Prints: true,false,false |
| f. | Prints: true,false,true |
| g. | Prints: true,true,false |
| h. | Prints: true,true,true |
| i. | None of the above |

**Question 3**

import java.util.\*;

class GFC112 {

public static void main (String[] args) {

Object m = new LinkedHashSet();

System.out.print((m instanceof Collection)+",");

System.out.print((m instanceof Set)+",");

System.out.print(m instanceof List);

}}

What is the result of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | Prints: false,false,false |
| b. | Prints: false,false,true |
| c. | Prints: false,true,false |
| d. | Prints: false,true,true |
| e. | Prints: true,false,false |
| f. | Prints: true,false,true |
| g. | Prints: true,true,false |
| h. | Prints: true,true,true |
| i. | None of the above |

**Question 4**

import java.util.\*;

class GFC113 {

public static void main (String[] args) {

Object m = new LinkedHashSet();

System.out.print((m instanceof Collection)+",");

System.out.print((m instanceof Set)+",");

System.out.print(m instanceof HashSet);

}}

What is the result of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | Prints: false,false,false |
| b. | Prints: false,false,true |
| c. | Prints: false,true,false |
| d. | Prints: false,true,true |
| e. | Prints: true,false,false |
| f. | Prints: true,false,true |
| g. | Prints: true,true,false |
| h. | Prints: true,true,true |
| i. | None of the above |

**Question 5**

import java.util.\*;

class GFC116 {

public static void main (String[] args) {

Object x = new Vector().elements();

System.out.print((x instanceof Enumeration)+",");

System.out.print((x instanceof Iterator)+",");

System.out.print(x instanceof ListIterator);

}}

What is the result of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | Prints: false,false,false |
| b. | Prints: false,false,true |
| c. | Prints: false,true,false |
| d. | Prints: false,true,true |
| e. | Prints: true,false,false |
| f. | Prints: true,false,true |
| g. | Prints: true,true,false |
| h. | Prints: true,true,true |
| i. | None of the above |

**Question 6**

import java.util.\*;

class GFC117 {

public static void main (String[] args) {

Object i1 = new HashMap(), i2 = new TreeMap();

System.out.print((i1 instanceof SortedMap)+",");

System.out.print((i2 instanceof SortedMap)+",");

System.out.print(i1 instanceof Collection);

}}

What is the result of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | Prints: false,false,false |
| b. | Prints: false,false,true |
| c. | Prints: false,true,false |
| d. | Prints: false,true,true |
| e. | Prints: true,false,false |
| f. | Prints: true,false,true |
| g. | Prints: true,true,false |
| h. | Prints: true,true,true |
| i. | None of the above |

**Question 7**

import java.util.\*;

class GFC118 {

public static void main (String[] args) {

Object i = new ArrayList().listIterator();

System.out.print((i instanceof List)+",");

System.out.print((i instanceof Iterator)+",");

System.out.print(i instanceof ListIterator);

}}

What is the result of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | Prints: false,false,false |
| b. | Prints: false,false,true |
| c. | Prints: false,true,false |
| d. | Prints: false,true,true |
| e. | Prints: true,false,false |
| f. | Prints: true,false,true |
| g. | Prints: true,true,false |
| h. | Prints: true,true,true |
| i. | None of the above |

**Question 8**

Which of the following classes allow elements to be accessed in the order that they were added?

|  |  |
| --- | --- |
| a. | HashMap |
| b. | HashSet |
| c. | Hashtable |
| d. | TreeMap |
| e. | TreeSet |
| f. | LinkedHashMap |
| g. | LinkedHashSet |

**Question 9**

Which of the following are true statements?

|  |  |
| --- | --- |
| a. | The Enumeration interface was introduced with the collections framework with Java 1.2. |
| b. | The Enumeration interface declares only two methods: hasMoreElements and nextElement. |
| c. | The Iterator interface extends the Enumeration interface. |
| d. | The Iterator interface declares a total of three methods. |

**Question 10**

Which of the following is a true statement?

|  |  |
| --- | --- |
| a. | The Iterator interface declares only two methods: hasMoreElements and nextElement. |
| b. | The ListIterator interface extends both the List and Iterator interfaces. |
| c. | The ListIterator interface was introduced with Java 1.2 to replace the older Iterator interface that was released with Java 1.0. |
| d. | The ListIterator interface declares only three methods: hasNext, next and remove. |
| e. | None of the above. |

**Question 11**

|  |  |  |
| --- | --- | --- |
|  | • | Stores key/value pairs. |
|  | • | Allows null elements, keys, and values. |
|  | • | Duplicate entries replace old entries. |
|  | • | Entries are not sorted using a Comparator or the Comparable interface. |
|  | • | The iteration order is unspecified. |

Which of these classes provides the specified features?

|  |  |
| --- | --- |
| a. | LinkedList |
| b. | LinkedHashMap |
| c. | LinkedHashSet |
| d. | TreeMap |
| e. | TreeSet |
| f. | HashMap |
| g. | HashSet |
| h. | Hashtable |
| i. | None of the above |

**Question 12**

|  |  |  |
| --- | --- | --- |
|  | • | Stores key/value pairs. |
|  | • | Does not allow null elements, keys, and values. |

Which of these classes provides the specified features?

|  |  |
| --- | --- |
| a. | LinkedList |
| b. | LinkedHashMap |
| c. | LinkedHashSet |
| d. | TreeMap |
| e. | TreeSet |
| f. | HashMap |
| g. | HashSet |
| h. | Hashtable |
| i. | None of the above |

**Question 13**

|  |  |  |
| --- | --- | --- |
|  | • | Stores key/value pairs. |
|  | • | Duplicate entries replace old entries. |
|  | • | Provides constant-time performance for the add, contains and remove operations. |

Which of these classes provides the specified features?

|  |  |
| --- | --- |
| a. | LinkedHashMap |
| b. | LinkedHashSet |
| c. | LinkedList |
| d. | TreeMap |
| e. | TreeSet |
| f. | HashMap |
| g. | HashSet |
| h. | Hashtable |

**Question 14**

import java.util.\*;

class GFC114 {

public static void main (String[] args) {

Object a = new ArrayList();

System.out.print((a instanceof Collections)+",");

System.out.print((a instanceof Arrays)+",");

System.out.print(a instanceof List);

}}

What is the result of attempting to compile and run the program?

|  |  |
| --- | --- |
| a. | Prints: false,false,false |
| b. | Prints: false,false,true |
| c. | Prints: false,true,false |
| d. | Prints: false,true,true |
| e. | Prints: true,false,false |
| f. | Prints: true,false,true |
| g. | Prints: true,true,false |
| h. | Prints: true,true,true |
| i. | None of the above |

**Question 15**

|  |  |  |
| --- | --- | --- |
|  | • | Each element must be unique. |
|  | • | Contains no duplicate elements. |
|  | • | Elements are not key/value pairs. |
|  | • | Entries are not sorted using a Comparator or the Comparable interface. |
|  | • | The iteration order is determined by the insertion order. |

Which of these classes provides the specified features?

|  |  |
| --- | --- |
| a. | HashMap |
| b. | HashSet |
| c. | Hashtable |
| d. | LinkedHashMap |
| e. | LinkedHashSet |
| f. | LinkedList |
| g. | TreeMap |
| h. | TreeSet |
| i. | None of the above |

**Question 16**

Suppose that you would like to create a new instance of a class that implements the Set interface, and you would like the new instance to be initialized with the elements of an existing Set. If you would like the iteration order of the new Set to be the same as that of the existing Set, then which concrete implementation of the Set interface should be used for the new instance?

|  |  |
| --- | --- |
| a. | Hashtable |
| b. | HashMap |
| c. | HashSet |
| d. | LinkedHashSet |
| e. | TreeMap |
| f. | TreeSet |
| g. | None of the above. |

**Question 17**

Suppose that you would like to create a new instance of a class that implements the Map interface, and you would like the new instance to be initialized with the elements of an existing Map. If you would like the iteration order of the new Map to be the same as that of the existing Map, then which concrete implementation of the Map interface should be used for the new instance?

|  |  |
| --- | --- |
| a. | Hashtable |
| b. | HashSet |
| c. | HashMap |
| d. | TreeMap |
| e. | TreeSet |
| f. | LinkedHashMap |
| g. | None of the above. |

**Question 18**

|  |  |  |
| --- | --- | --- |
|  | • | Stores key/value pairs. |
|  | • | Allows null elements, keys, and values. |
|  | • | Duplicate entries replace old entries. |
|  | • | The least recently used element can be removed automatically when a new element is added. |

Which of these classes provides the specified features?

|  |  |
| --- | --- |
| a. | LinkedHashMap |
| b. | LinkedHashSet |
| c. | LinkedList |
| d. | TreeMap |
| e. | TreeSet |
| f. | HashMap |
| g. | HashSet |
| h. | Hashtable |
| i. | None of the above |

**Question 19**

Which of the following classes would provide the most efficient implementation of a First In First Out queue?

|  |  |
| --- | --- |
| a. | ArrayList |
| b. | LinkedHashMap |
| c. | LinkedHashSet |
| d. | LinkedList |
| e. | HashMap |
| f. | HashSet |
| g. | Hashtable |
| h. | TreeMap |
| i. | TreeSet |
| j. | Vector |
| k. | None of the above |

**Question 20**

In addition to implementing the List interface, which of the following also provides methods to get, add and remove elements from the head and tail of the list without specifying an index?

|  |  |
| --- | --- |
| a. | Collection |
| b. | ArrayList |
| c. | LinkedList |
| d. | List |
| e. | Vector |
| f. | None of the above |

**Question 21**

Which of the following is a true statement?

|  |  |
| --- | --- |
| a. | All implementations of the List interface provide fast random access. |
| b. | A LinkedList provides faster random access than an ArrayList. |
| c. | The LinkedList implements the RandomAccess interface. |
| d. | Each collection that implements the List interface must also implement the RandomAccess interface. |
| e. | The RandomAccess interface declares methods that use of an index argument to access elements of the List. |
| f. | The RandomAccess interface declares the next and hasNext methods. |
| g. | None of the above. |

**Question 22**

Which of the following is not a true statement?

|  |  |
| --- | --- |
| a. | The Iterator interface declares only three methods: hasNext, next and remove. |
| b. | The ListIterator interface extends both the List and Iterator interfaces. |
| c. | The ListIterator interface provides forward and backward iteration capabilities. |
| d. | The ListIterator interface provides the ability to modify the List during iteration. |
| e. | The ListIterator interface provides the ability to determine its position in the List. |
| f. | None of the above. |