Suppose that you would like to create an instance of a new Map that has an iteration order that is the same as the iteration order of an existing instance of a Map. Which concrete implementation of the Map interface should be used for the new instance?

1. TreeMap
2. HashMap
3. LinkedHashMap
4. The answer depends on the implementation of the existing instance

Ans:--

Which statement is true for the class java.util.HashSet?

1. The elements in the collection are ordered.
2. The collection is guaranteed to be immutable.
3. The elements in the collection are guaranteed to be unique.
4. The elements in the collection are accessed using a unique key.

Ans:---

Which statement is true for the class java.util.HashSet?

1. The elements in the collection are ordered.
2. The collection is guaranteed to be immutable.
3. The elements in the collection are guaranteed to be unique.
4. The elements in the collection are accessed using a unique key.

Ans:---

You need to store elements in a collection that guarantees that no duplicates are stored. Which one of the following interfaces provide that capability?

1. Java.util.Map
2. Java.util.List
3. Java.util.Collection
4. None of the above

Ans:----

You need to store elements in a collection that guarantees that no duplicates are stored and all elements can be accessed in natural order. Which interface provides that capability?

1. java.util.Map
2. java.util.Set
3. java.util.List
4. java.util.Collection

Ans:------

What will be the output of the program?

import java.util.\*;

class I

{

public static void main (String[] args)

{

Iterator i = new ArrayList().iterator();

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

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

System.out.print(i instanceof ListIterator);

}

}

1. Prints: false, false, false
2. Prints: false, false, true
3. Prints: false, true, false
4. Prints: false, true, true

Ans:---

Which two statements are true about comparing two instances of the same class, given that the equals() and hashCode() methods have been properly overridden?

1. If the equals() method returns true, the hashCode() comparison == must return true.
2. If the equals() method returns false, the hashCode() comparison != must return true.
3. If the hashCode() comparison == returns true, the equals() method must return true.
4. If the hashCode() comparison == returns true, the equals() method might return true.
5. 1 and 4
6. 2 and 3
7. 3 and 4
8. 1 and 3

Ans:----

Which interface does java.util.HashTable implement?

1. Java.util.Map
2. Java.util.List
3. Java.util.HashTable
4. Java.util.Collection

Ans:---

What will be the output of the program?

import java.util.\*;

class H

{

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);

}

}

1. Prints: false,false,false
2. Prints: false,false,true
3. Prints: false,true,false
4. Prints: true,false,false

Ans:----