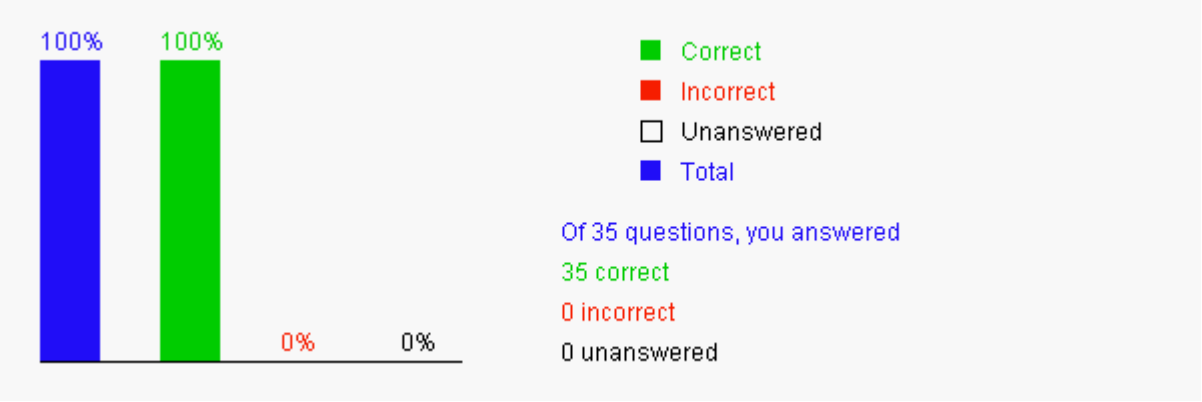
This quiz is for students to practice. A large number of additional quiz is available for instructors using Quiz Generator from the Instructor's Resource Website.
Videos for Java, Python, and C++ can be found at https://yongdanielliang.github.io/revelvideos.html.

Chapter 21 Sets and Maps



Please send suggestions and errata to Dr. Liang at y.daniel.liang@gmail.com. Indicate which book and edition you are using. Thanks!

Section 21.2 Sets

21.1 Which of the data types below does not allow duplicates?

- (●) A. Set
- ( ) B. List
- ( ) C. Vector
- ( ) D. Stack
- ( ) E. LinkedList

Your answer is correct ✔

Section 21.2.3 TreeSet

21.2 Which of the data types below could be used to store elements in their natural order based on the compareTo method?

- ( ) A. HashSet
- (●) B. TreeSet
- ( ) C. LinkedHashSet
- ( ) D. Collection
- ( ) E. Set

Your answer is correct ✔

Section 21.2.1 HashSet

21.3 What is the output for the following code?

```
import java.util.*;
public class Test {
  public static void main(String[] args) {
    Set<A> set = new HashSet<A>();
    set.add(new A());
    set.add(new A());
    set.add(new A());
    set.add(new A());
    System.out.println(set);
  }
}

class A  {
  int r = 1;

  public String toString() {
    return r + "";
  }

  public boolean equals(Object o) {
    return this.r == ((A)o).r;
  }

  public int hashCode() {
    return r;
  }
}
```

- (●) A. [1]

B. [1, 1]
C. [1, 1, 1]
D. [1, 1, 1, 1]

Your answer is correct 

21.4 If two objects o1 and o2 are equal, what are the values for o1.equals(o2) and o1.hashCode() == o2.hashCode()?

A. true true
B. true false
C. false true
D. false false

Your answer is correct 

Section 21.3 Comparing the Performance of Sets and Lists

21.5 What is the output of the following code?

```
import java.util.*;

import java.util.*;

public class Test {
  public static void main(String[] args) {
    Set<String> set1 = new HashSet<>();
    set1.add("Atlanta");
    set1.add("Macon");
    set1.add("Savanna");

    Set<String> set2 = new HashSet<>();
    set2.add("Atlanta");
    set2.add("Macon");
    set2.add("Savanna");

    Set<String> set3 = new HashSet<>();
    set3.add("Macon");
    set3.add("Savanna");
    set3.add("Atlanta");

    System.out.println(set1.equals(set2) + " " + set1.equals(set3));
  }
}
```

A. true true
B. true false
C. false false
D. false true

Your answer is correct 

21.6 What is the output for the following code?

```
import java.util.*;
public class Test {
  public static void main(String[] args) {
    Set<A> set = new HashSet<>();
    set.add(new A());
    set.add(new A());
    set.add(new A());
    set.add(new A());
    System.out.println(set);
  }
}

class A  {
  int r = 1;

  public String toString() {
    return r + "";
  }

  public int hashCode() {
    return r;
  }
}
```

A. [1]
B. [1, 1]
C. [1, 1, 1]

- (●) D. [1, 1, 1, 1]

Your answer is correct

21.7 What is the output for the following code?

```
import java.util.*;
public class Test {
  public static void main(String[] args) {
    Set<A> set = new HashSet<>();
    set.add(new A());
    set.add(new A());
    set.add(new A());
    set.add(new A());
    System.out.println(set);
  }
}

class A  {
  int r = 1;

  public String toString() {
    return r + "";
  }

  public boolean equals(Object o) {
    return this.r == ((A)o).r;
  }
}
```

- ( ) A. [1]
- ( ) B. [1, 1]
- ( ) C. [1, 1, 1]
- (●) D. [1, 1, 1, 1]

Your answer is correct

21.8 Which of the following data types have iterators?

- [x] A. HashSet
- [x] B. TreeSet
- [x] C. ArrayList
- [x] D. LinkedList
- [x] E. LinkedHashSet

Your answer is correct
Explanation: The Collection interface has the iterator() method to return an iterator from a collection.

21.9 To get an iterator from a set, you may use the _____ method.

- ( ) A. getIterator
- ( ) B. findIterator
- (●) C. iterator
- ( ) D. iterators

Your answer is correct

Section 21.4 Case Study: Counting Keywords

21.10 Suppose set s1 is [1, 2, 5] and set s2 is [2, 3, 6]. After s1.addAll(s2), s1 is _____.

- ( ) A. [1, 2, 2, 3, 5, 6]
- (●) B. [1, 2, 3, 5, 6]
- ( ) C. [1, 5]
- ( ) D. [2]

Your answer is correct

21.11 Suppose set s1 is [1, 2, 5] and set s2 is [2, 3, 6]. After s1.addAll(s2), s2 is _____.

- ( ) A. [1, 2, 2, 3, 5, 6]
- ( ) B. [1, 2, 3, 5, 6]
- ( ) C. [1, 5]
- (●) D. [2, 3, 6]
- ( ) E. [2]

Your answer is correct

21.12 Suppose set s1 is [1, 2, 5] and set s2 is [2, 3, 6]. After s1.removeAll(s2), s1 is _____.

- A. [1, 2, 2, 3, 5, 6]
- B. [1, 2, 3, 5, 6]
- C. [1, 5]
- D. [2]

Your answer is correct

21.13 Suppose set s1 is [1, 2, 5] and set s2 is [2, 3, 6]. After s1.retainAll(s2), s1 is _____.

- A. [1, 2, 2, 3, 5, 6]
- B. [1, 2, 3, 5, 6]
- C. [1, 5]
- D. [2]

Your answer is correct

21.14 The output of the following code is _____.

```
LinkedHashSet<String> set1 = new LinkedHashSet<>();
set1.add("New York");
LinkedHashSet<String> set2 = (LinkedHashSet<String>)(set1.clone());
set1.add("Atlanta");
set2.add("Dallas");
System.out.println(set2);
```

- A. [New York]
- B. [New York, Atlanta]
- C. [New York, Atlanta, Dallas]
- D. [New York, Dallas]

Your answer is correct

21.15 The output of the following code is _____.

```
LinkedHashSet<String> set1 = new LinkedHashSet<>();
set1.add("New York");
LinkedHashSet<String> set2 = set1;
set1.add("Atlanta");
set2.add("Dallas");
System.out.println(set2);
```

- A. [New York]
- B. [New York, Atlanta]
- C. [New York, Atlanta, Dallas]
- D. [New York, Dallas]

Your answer is correct

21.16 Analyze the following code:

```
import java.util.*;

public class Test {
  public static void main(String[] args) {
    HashSet<String> set1 = new HashSet<>();
    set1.add("red");
    Set<String> set2 = set1.clone();
  }
}
```

- [ ] A. Line 5 is wrong because a HashSet object cannot be cloned.
- [x] B. Line 5 has a compile error because set1.clone() returns an Object. You have to cast it to Set in order to compile it.
- [ ] C. The program will be fine if set1.clone() is replaced by (Set<String>)set1.clone()
- [x] D. The program will be fine if set1.clone() is replaced by (Set<String>)(set1.clone())
- [x] E. The program will be fine if set1.clone() is replaced by (HashSet<String>)(set1.clone())

Your answer is correct

21.17 Analyze the following code:

```
import java.util.*;
```

```
public class Test {
  public static void main(String[] args) {
    Set<String> set1 = new HashSet<>();
    set1.add("red");
    Set set2 = set1.clone();
  }
}
```

- [x] A. Line 5 is wrong because the declared type for set1 is Set and the clone method is not defined Set.
- [ ] B. The program will be fine if set1.clone() is replaced by (HashSet)set1.clone()
- [x] C. The program will be fine if set1.clone() is replaced by (Set)((HashSet)set1).clone()
- [x] D. The program will be fine if set1.clone() is replaced by (HashSet)((HashSet)set1).clone()
- [ ] E. The program will be fine if set1.clone() is replaced by (LinkedHashSet)((HashSet)set1).clone()

Your answer is correct

Explanation: For (E), the program will compile fine, but will get a runtime ClassCastException because set1 is a HashSet, not a LinkedHashSet.

21.18 If you want to store non-duplicated objects in the order in which they are inserted, you should use _____.

- ( ) A. HashSet
- (•) B. LinkedHashSet
- ( ) C. TreeSet
- ( ) D. ArrayList
- ( ) E. LinkedList

Your answer is correct

21.19 Which of the following statements are true?

- [x] A. All the methods in HashSet are inherited from the Collection interface.
- [ ] B. All the methods in TreeSet are inherited from the Collection interface.
- [x] C. All the methods in LinkedHashSet are inherited from the Collection interface.
- [x] D. All the methods in Set are inherited from the Collection interface.
- [x] E. All the concrete classes of Collection have at least two constructors. One is the no-arg constructor that constructs an empty collection. The other constructs instances from a collection.

Your answer is correct

Explanation: TreeSet has the first(), last(), headSet(toElement), and tailSet(fromElement) methods.

21.20 Which of the following is correct to perform the set union of two sets s1 and s2?

- ( ) A. s1.union(s2)
- ( ) B. s1 + s2
- (•) C. s1.addAll(s2)
- ( ) D. s1.add(s2)

Your answer is correct

21.21 Which of the following is correct to perform the set difference of two sets s1 and s2?

- ( ) A. s1.difference(s2)
- ( ) B. s1 - s2
- ( ) C. s1.subtract(s2)
- (•) D. s1.removeAll(s2)

Your answer is correct

21.22 Which of the following is correct to perform the set intersection of two sets s1 and s2?

- ( ) A. s1.intersect(s2)
- ( ) B. s1.join(s2)
- (•) C. s1.retainAll(s2)
- ( ) D. s1.intersection(s2)

Your answer is correct

21.23 Analyze the following code.

```
import java.util.*;

public class Test {
```

```
  public static void main(String[] args) throws Exception {
    Set<String> set = new TreeSet<>();

    set.add("Red");
    set.add("Green");
    set.add("Blue");

    System.out.println(set.first());
  }
}
```

- ○ A. The program displays Red
- ○ B. The program displays Blue
- ○ C. The program displays Green
- ○ D. The program may display Red, Blue, or Green.
- ◉ E. The program cannot compile, because the first() method is not defined in Set.

Your answer is correct 
Explanation: first() is defined in TreeSet. To compile this program, replace Set set = new TreeSet() with TreeSet set = new TreeSet().

21.24 Analyze the following code.

```
import java.util.*;

public class Test {
  public static void main(String[] args) throws Exception {
    TreeSet<String> set = new TreeSet<>();

    set.add("Red");
    set.add("Green");
    set.add("Blue");

    System.out.println(set.last());
  }
}
```

- ◉ A. The program displays Red
- ○ B. The program displays Blue
- ○ C. The program displays Green
- ○ D. The program may display Red, Blue, or Green.
- ○ E. The program cannot compile, because the last() method is not defined in Set.

Your answer is correct 

21.25 Analyze the following code.

```
import java.util.*;

public class Test {
  public static void main(String[] args) throws Exception {
    TreeSet<String> set = new TreeSet<>();

    set.add("Red");
    set.add("Yellow");
    set.add("Green");
    set.add("Blue");
    SortedSet temp = set.headSet("Purple");

    System.out.println(temp.first());
  }
}
```

- ○ A. The program displays Red
- ◉ B. The program displays Blue
- ○ C. The program displays Green
- ○ D. The program displays Yellow
- ○ E. The program displays Purple

Your answer is correct 

21.26 Analyze the following code.

```
import java.util.*;

public class Test {
  public static void main(String[] args) throws Exception {
    TreeSet<String> set = new TreeSet<>();

    set.add("Red");
```

```
    set.add("Yellow");
    set.add("Green");
    set.add("Blue");
    SortedSet temp = set.tailSet("Purple");

    System.out.println(temp.first());
  }
}
```

- (●) A. The program displays Red
- ( ) B. The program displays Blue
- ( ) C. The program displays Green
- ( ) D. The program displays Yellow
- ( ) E. The program displays Purple

Your answer is correct

Section 21.5 Maps

21.27 Analyze the following code:

```
public class Test {
  public static void main(String[] args) {
    Map<String, String> map = new HashMap<>();
    map.put("123", "John Smith");
    map.put("111", "George Smith");
    map.put("123", "Steve Yao");
    map.put("222", "Steve Yao");
  }
}
```

- ( ) A. After all the four entries are added to the map, "123" is a key that corresponds to the value "John Smith".
- (●) B. After all the four entries are added to the map, "123" is a key that corresponds to the value "Steve Yao".
- ( ) C. After all the four entries are added to the map, "Steve Yao" is a key that corresponds to the value "222".
- ( ) D. After all the four entries are added to the map, "John Smith" is a key that corresponds to the value "123".
- ( ) E. A runtime error occurs because two entries with the same key "123" are added to the map.

Your answer is correct
Explanation: The signature of the put method is put(key, value). So the first parameter in the put method is the key. When a new entry with the same key is added to the map, the existing entry with the same key is replaced by the new entry.

21.28 To empty a Collection or a Map, you use the _____ method.

- ( ) A. empty
- (●) B. clear
- ( ) C. zero
- ( ) D. setEmpty

Your answer is correct

21.29 The Collection interface is the base interface for _____.

- [x] A. Set
- [x] B. List
- [x] C. ArrayList
- [x] D. LinkedList
- [ ] E. Map

Your answer is correct
Explanation: The Collection is not the base interface for Map.

21.30 The elements in _____ are sorted.

- [x] A. TreeSet
- [ ] B. List
- [x] C. TreeMap
- [ ] D. HashSet
- [ ] E. LinkedHashSet

Your answer is correct

21.31 Suppose your program frequently tests whether a student is in a soccer team, what is the best data structure to store the students in a soccer team?

- ( ) A. ArrayList

- (●) B. HashSet
- ( ) C. LinkedList
- ( ) E. Vector

Your answer is correct 

21.32 Suppose your program frequently tests whether a student is in a soccer team and also need to know the student?s information such as phone number, address, and age, what is the best data structure to store the students in a soccer team?

- ( ) A. ArrayList
- (●) B. HashMap
- ( ) C. TreeMap
- ( ) D. LinkedList
- ( ) E. HashSet

Your answer is correct 

21.33 The Map is the base interface for _____.

- [x] A. TreeMap
- [x] B. HashMap
- [x] C. LinkedHashMap
- [ ] D. ArrayList
- [ ] E. LinkedList

Your answer is correct 

21.34 Which of the following are correct methods in Map?

- [x] A. put(Object key, Object value)
- [ ] B. put(Object value, Object key)
- [x] C. get(Object key)
- [ ] D. get(int index)

Your answer is correct 

21.35 Which of the following are correct methods in Map?

- [x] A. containsKey(Object key)
- [x] B. containsValue(Object value)
- [x] C. remove(Object key)
- [ ] D. remove(int index)
- [x] E. isEmpty()

Your answer is correct