

DURGA ONLINE EXAMS

DURGAJOBS.COM

Continuous Job Updates For Every hour

Test Your Knowledge

[HOME](#)241) **Given:**

```
10. abstract public class Employee {  
11.     protected abstract double getSalesAmount();  
12.     public double getCommision() {  
13.         return getSalesAmount() * 0.15;  
14.     }  
15. }  
16. class Sales extends Employee {  
17.     // insert method here  
18. }
```

Which two methods, inserted independently at line 17, correctly complete the Sales class?
(Choose two.)

- 1) `double getSalesAmount() { return 1230.45; }`
- 2) `public double getSalesAmount() { return 1230.45; }`
- 3) `private double getSalesAmount() { return 1230.45; }`
- 4) `protected double getSalesAmount() { return 1230.45; }`

Your Selected options :: none ❌

Correct Options :: 2, 4

[Click Here for Explanation](#)242) **Given:**

```
1. public class Drink implements Comparable {  
2.     public String name;  
3.     public int compareTo(Object o) {  
4.         return 0;  
5.     }  
6. }
```

and:

```
20. Drink one = new Drink();  
21. Drink two = new Drink();  
22. one.name= "Coffee";  
23. two.name= "Tea";  
23. TreeSet set = new TreeSet();  
24. set.add(one);  
25. set.add(two);
```

A programmer iterates over the TreeSet and prints the name of each Drink object.
What is the result?

- 1) Tea
- 2) Coffee
- 3) Coffee Tea
- 4) Compilation fails.
- 5) The code runs with no output.
- 6) An exception is thrown at runtime.

Your Selected options :: none ❌

Correct Options :: 2

[Click Here for Explanation](#)243) **Given:**

```
34. HashMap props = new HashMap();  
35. props.put("key45", "some value");  
36. props.put("key12", "some other value");  
37. props.put("key39", "yet another value");  
38. Set s = props.keySet();  
39. // insert code here
```

What, inserted at line 39, will sort the keys in the props HashMap?

- 1) `Arrays.sort(s);`

- 2) `s = new TreeSet(s);`
- 3) `Collections.sort(s);`
- 4) `s = new SortedSet(s);`

Your Selected options :: none ❌

Correct Options :: 2

[Click Here for Explanation](#)

244) Given:

```
1. import java.util.*;
2. public class Test {
3.     public static void main(String[] args) {
4.         List<String> strings = new ArrayList<String>();
5.         // insert code here
6.     }
7. }
```

Which four, inserted at line 5, will allow compilation to succeed? (Choose four.)

- 1) `String s = strings.get(0);`
- 2) `Iterator i1 = strings.iterator();`
- 3) `String[] array1 = strings.toArray();`
- 4) `Iterator<String> i2 = strings.iterator();`
- 5) `String[] array2 = strings.toArray(new String[1]);`
- 6) `Iterator<String> i3 = strings.iterator<String>();`

Your Selected options :: none ❌

Correct Options :: 1, 2, 4, 5

[Click Here for Explanation](#)

245) Given:

```
1. public class Test {
2.     public <T extends Comparable> T findLarger(T x, T y) {
3.         if(x.compareTo(y) > 0) {
4.             return x;
5.         } else {
6.             return y;
7.         }
8.     }
9. }
and:
```

22. `Test t = new Test();`

23. // insert code here

Which two will compile without errors when inserted at line 23? (Choose two.)

- 1) `Object x = t.findLarger(123, "456");`
- 2) `int x = t.findLarger(123, new Double(456));`
- 3) `int x = t.findLarger(123, new Integer(456));`
- 4) `int x = (int) t.findLarger(new Double(123), new Double(456));`

Your Selected options :: none ❌

Correct Options :: 1, 3

[Click Here for Explanation](#)

246) Given:

```
5. import java.util.*;
6. public class SortOf {
7.     public static void main(String[] args) {
8.         ArrayList<Integer> a = new ArrayList<Integer>();
9.         a.add(1); a.add(5); a.add(3);
10.        Collections.sort(a);
11.        a.add(2);
12.        Collections.reverse(a);
13.        System.out.println(a);
14.    }
15. }
```

What is the result?

- 1) `[1, 2, 3, 5]`
- 2) `[2, 1, 3, 5]`

- 3) [2, 5, 3, 1]
- 4) [5, 3, 2, 1]
- 5) [1, 3, 5, 2]
- 6) **Compilation fails.**
- 7) **An exception is thrown at runtime.**

Your Selected options :: none ❌

Correct Options :: 3

[Click Here for Explanation](#)

- 247) **A programmer has an algorithm that requires a java.util.List that provides an efficient implementation of add(0, object), but does NOT need to support quick random access. What supports these requirements?**

- 1) **java.util.Queue**
- 2) **java.util.ArrayList**
- 3) **java.util.LinearList**
- 4) **java.util.LinkedList**

Your Selected options :: none ❌

Correct Options :: 4

[Click Here for Explanation](#)

- 248) **Given:**

```
12. import java.util.*;
13. public class Explorer1 {
14. public static void main(String[] args) {
15. TreeSet<Integer> s = new TreeSet<Integer>();
16. TreeSet<Integer> subs = new TreeSet<Integer>();
17. for(int i = 606; i < 613; i++) 18. if(i%2 == 0) s.add(i);
19. subs = (TreeSet)s.subSet(608, true, 611, true);
20. s.add(609);
21. System.out.println(s + " " + subs);
22. }
23. }
```

What is the result?

- 1) **Compilation fails.**
- 2) **An exception is thrown at runtime.**
- 3) **[608, 609, 610, 612] [608, 610]**
- 4) **[608, 609, 610, 612] [608, 609, 610]**
- 5) **[606, 608, 609, 610, 612] [608, 610]**
- 6) **[606, 608, 609, 610, 612] [608, 609, 610]**

Your Selected options :: none ❌

Correct Options :: 6

[Click Here for Explanation](#)

- 249) **Given a class whose instances, when found in a collection of objects, are sorted by using the compareTo() method, which two statements are true? (Choose two.)**

- 1) **The class implements java.lang.Comparable.**
- 2) **The class implements java.util.Comparator.**
- 3) **The interface used to implement sorting allows this class to define only one sort sequence.**
- 4) **The interface used to implement sorting allows this class to define many different sort sequences.**

Your Selected options :: none ❌

Correct Options :: 1, 3

[Click Here for Explanation](#)

250) **Given:**

```

3. import java.util.*;
4. public class Mapit {
5. public static void main(String[] args) {
6. Set<Integer> set = new HashSet<Integer>();
7. Integer i1 = 45;
8. Integer i2 = 46;
9. set.add(i1);
10. set.add(i1);
11. set.add(i2); System.out.print(set.size() + " ");
12. set.remove(i1); System.out.print(set.size() + " ");
13. i2 = 47;
14. set.remove(i2); System.out.print(set.size() + " ");
15. }
16. }

```

What is the result?

- 1) **2 1 0**
- 2) **2 1 1**
- 3) **3 2 1**
- 4) **3 2 2**
- 5) **Compilation fails.**
- 6) **An exception is thrown at runtime.**

Your Selected options :: none ❌**Correct Options :: 2**[Click Here for Explanation](#)

[« Prev](#) | [1](#) | [2](#) | [3](#) | [4](#) | [5](#) | [6](#) | [7](#) | [8](#) | [9](#) | [10](#) | [11](#) | [12](#) | [13](#) | [14](#) | [15](#) |
[16](#) | [17](#) | [18](#) | [19](#) | [20](#) | [21](#) | [22](#) | [23](#) | [24](#) | [25](#) | [26](#) | [27](#) | [28](#) | [29](#) | [30](#) |
[Next »](#)

Total No.of Questions	:: 292
Total No.of Answered Questions	:: 0
Total No.of Unanswered Questions	:: 292
Marks	:: 0/292(0%)

feedback :: feedback@durgajobs.com

© durgajobs.com All Rights Reserved