# DURGA ONLINE EXAMS

## Test Your Knowledge

---

**HOME**

---

91) **Given:**
   **1. public class Donkey2 {**
   **2. public static void main(String[] args) {**
   **3. boolean assertsOn = true;**
   **4. assert (assertsOn) : assertsOn = true;**
   **5. if(assertsOn) {**
   **6. System.out.println("assert is on");**
   **7. }**
   **8. }**
   **9. }**
   **If class Donkey is invoked twice, the first time without assertions enabled, and the second time with assertions enabled, what are the results?**

   1) **no output**

   2) **no output**
      **assert is on**

   3) **assert is on**

   4) **no output**
      **An AssertionError is thrown.**

   5) **assert is on**
      **An AssertionError is thrown.**

   **Your Selected options  ::   none**  ✖

   **Correct Options          ::   3**

   [ Click Here for Explanation ]

---

92) **Given:**
   **8. public class test {**
   **9. public static void main(String [] a) {**
   **10. assert a.length == 1;**
   **11. }**
   **12. }**
   **Which two will produce an AssertionError? (Choose two.)**

   1) **java test**

   2) **java -ea test**

   3) **java test file1**

   4) **java -ea test file1**

   5) **java -ea test file1 file2**

   6) **java -ea:test test file1**

   **Your Selected options  ::   none**  ✖

   **Correct Options          ::   2, 5**

   [ Click Here for Explanation ]

---

93) **Click the Exhibit button.**
   **1. public class Test {**
   **2.**
   **3. public static void main(String [] args) {**
   **4. boolean assert = true;**
   **5. if(assert) {**
   **6. System.out.println(â€assert is trueâ€);**
   **7. }**
   **8. }**
   **9.**
   **10. }**
   **Given:**
       **javac -source 1.3 Test.java**
   **What is the result?**

   1) **Compilation fails.**

2) **Compilation succeeds with errors.**

3) **Compilation succeeds with warnings.**

4) **Compilation succeeds without warnings or errors.**

> **Your Selected options  ::  none**  ✖
>
> **Correct Options           ::  3**

> Click Here for Explanation

94) **Given:**
    **1. public class Donkey {**
    **2. public static void main(String[] args) {**
    **3. boolean assertsOn = false;**
    **4. assert (assertsOn) : assertsOn = true;**
    **5. if(assertsOn) {**
    **6. System.out.println("assert is on");**
    **7. }**
    **8. }**
    **9. }**
    **If class Donkey is invoked twice, the first time without assertions enabled, and the second time with assertions enabled, what are the results?**

   1) **no output**

   2) **no output**
    **assert is on**

   3) **assert is on**

   4) **no output**
    **An AssertionError is thrown.**

   5) **assert is on**
    **An AssertionError is thrown.**

> **Your Selected options  ::  none**  ✖
>
> **Correct Options           ::  4**

> Click Here for Explanation

95) **Which statement is true?**

   1) **A class's finalize() method CANNOT be invoked explicitly.**

   2) **super.finalize() is called implicitly by any overriding finalize() method.**

   3) **The finalize() method for a given object is called no more than once by the garbage collector.**

   4) **The order in which finalize() is called on two objects is based on the order in which the two objects became finalizable.**

> **Your Selected options  ::  none**  ✖
>
> **Correct Options           ::  3**

> Click Here for Explanation

96) **Given:**
    **3. interface Animal { void makeNoise(); }**
    **4. class Horse implements Animal {**
    **5. Long weight = 1200L;**
    **6. public void makeNoise() { System.out.println("whinny"); }**
    **7. }**
    **8. public class Icelandic extends Horse {**
    **9. public void makeNoise() { System.out.println("vinny"); }**
    **10. public static void main(String[] args) {**
    **11. Icelandic i1 = new Icelandic();**
    **12. Icelandic i2 = new Icelandic();**
    **12. Icelandic i3 = new Icelandic();**
    **13. i3 = i1; i1 = i2; i2 = null; i3 = i1;**
    **14. }**
    **15. }**
    **When line 14 is reached, how many objects are eligible for the garbage collector?**

   1) **0**

   2) **1**

   3) **2**

   4) **3**

   5) **4**

6) **6**

**Your Selected options　::　none** ✖

**Correct Options　　　::　5**

```
Click Here for Explanation
```

97)　**Given:**
　　**1. public class GC {**
　　**2. private Object o;**
　　**3. private void doSomethingElse(Object obj) { o = obj; }**
　　**4. public void doSomething() {**
　　**5. Object o = new Object();**
　　**6. doSomethingElse(o);**
　　**7. o = new Object();**
　　**8. doSomethingElse(null);**
　　**9. o = null;**
　　**10. }**
　　**11. }**
　　**When the doSomething method is called, after which line does the Object created in line 5 become**
　　**available for garbage collection?**

　　　　1) **Line 5**

　　　　2) **Line 6**

　　　　3) **Line7**

　　　　4) **Line8**

　　　　5) **Line9**

　　　　6) **Line10**

**Your Selected options　::　none** ✖

**Correct Options　　　::　4**

```
Click Here for Explanation
```

98)　**Given:**
　　**11. public void genNumbers() {**
　　**12. ArrayList numbers = new ArrayList();**
　　**13. for (int i=0; i<10; i++) {**
　　**14. int value = i * ((int) Math.random());**
　　**15. Integer intObj = new Integer(value);**
　　**16. numbers.add(intObj);**
　　**17. }**
　　**18. System.out.println(numbers);**
　　**19. }**
　　**Which line of code marks the earliest point that an object referenced by intObj becomes a**
　　**candidate for garbage collection?**

　　　　1) **Line 16**

　　　　2) **Line 17**

　　　　3) **Line 18**

　　　　4) **Line 19**

　　　　5) **The object is NOT a candidate for garbage collection.**

**Your Selected options　::　none** ✖

**Correct Options　　　::　4**

```
Click Here for Explanation
```

99)　**Given:**
　　**11. class Snoochy {**
　　**12. Boochy booch;**
　　**13. public Snoochy() { booch = new Boochy(this); }**
　　**14. }**
　　**15.**
　　**16. class Boochy {**
　　**17. Snoochy snooch;**
　　**18. public Boochy(Snoochy s) { snooch = s; }**
　　**19. }**
　　**And the statements:**
　　**21. public static void main(String[] args) {**
　　**22. Snoochy snoog = new Snoochy();**
　　**23. snoog = null;**

**24. // more code here**
**25. }**
**Which statement is true about the objects referenced by snoog, snooch, and booch immediately after line 23 executes?**

1) **None of these objects are eligible for garbage collection.**

2) **Only the object referenced by booch is eligible for garbage collection.**

3) **Only the object referenced by snoog is eligible for garbage collection.**

4) **Only the object referenced by snooch is eligible for garbage collection.**

5) **The objects referenced by snooch and booch are eligible for garbage collection.**

**Your Selected options  ::  none**   ✖

**Correct Options         ::  5**

Click Here for Explanation

100)   **Which two are true? (Choose two.)**

1) **A finalizer may NOT be invoked explicitly.**

2) **The finalize method declared in class Object takes no action.**

3) **super.finalize() is called implicitly by any overriding finalize method.**

4) **The finalize method for a given object will be called no more than once by the garbage collector.**

5) **The order in which finalize will be called on two objects is based on the order in which the two objects became finalizable.**

**Your Selected options  ::  none**   ✖

**Correct Options         ::  2, 4**

Click Here for Explanation

| | | |
|---|---|---|
| **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**