DURGA ONLINE EXAMS

Test Your Knowledge

DURGAJOBS.COM
Continuous Job Updates For Every hour

HOME

---

161) **Given:**
    1. class SuperClass {
    2. public A getA() {
    3. return new A();
    4. }
    5. }
    6. class SubClass extends SuperClass {
    7. public B getA(){
    8. return new B();
    9. }
    10. }
Which statement is true?

    1) **Compilation will succeed if A extends B.**

    2) **Compilation will succeed if B extends A.**

    3) **Compilation will always fail because of an error in line 7.**

    4) **Compilation will always fail because of an error in line 8.**

    Your Selected options :: none ✖

    Correct Options   :: 2

Click Here for Explanation

---

162) **Given:**
    1. interface TestA { String toString(); }
    2. public class Test {
    3. public static void main(String[] args) {
    4. System.out.println(new TestA() {
    5. public String toString() { return "test"; }
    6. });
    7. }
    8. }
What is the result?

    1) **test**

    2) **null**

    3) **An exception is thrown at runtime.**

    4) **Compilation fails because of an error in line 1.**

    5) **Compilation fails because of an error in line 4.**

    6) **Compilation fails because of an error in line 5.**

    Your Selected options :: none ✖

    Correct Options   :: 1

Click Here for Explanation

---

163) **Given:**
    5. class Building { }
    6. public class Barn extends Building {
    7. public static void main(String[] args) {
    8. Building build1 = new Building();
    9. Barn barn1 = new Barn();
    10. Barn barn2 = (Barn) build1;
    11. Object obj1 = (Object) build1;
    12. String str1 = (String) build1;
    13. Building build2 = (Building) barn1;
    14. }
    15. }
Which is true?

    1) **If line 10 is removed, the compilation succeeds.**

    2) **If line 11 is removed, the compilation succeeds.**

    3) **If line 12 is removed, the compilation succeeds.**

4) **If line 13 is removed, the compilation succeeds.**

5) **More than one line must be removed for compilation to succeed.**

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

> Click Here for Explanation

---

164) **A team of programmers is involved in reviewing a proposed design for a new utility class. After some discussion, they realize that the current design allows other classes to access methods in the utility class that should be accessible only to methods within the utility class itself. What design issue has the team discovered?**

1) **Tight coupling**

2) **Low cohesion**

3) **High cohesion**

4) **Loose coupling**

5) **Weak encapsulation**

6) **Strong encapsulation**

> **Your Selected options :: none** ✖
>
> **Correct Options   :: 5**

> Click Here for Explanation

---

165) **Given:**
```
11. public class Yikes {
12.
13. public static void go(Long n) {System.out.println("Long ");}
14. public static void go(Short n) {System.out.println("Short ");}
15. public static void go(int n) {System.out.println("int ");}
16. public static void main(String [] args) {
17. short y = 6;
18. long z = 7;
19. go(y);
20. go(z);
21. }
22. }
```
What is the result?

1) **int Long**

2) **Short Long**

3) **Compilation fails.**

4) **An exception is thrown at runtime.**

> **Your Selected options :: none** ✖
>
> **Correct Options   :: 1**

> Click Here for Explanation

---

166)
```
10. import java.io.*;
11. class Animal {
12. Animal() { System.out.print("a"); }
13. }
14. class Dog extends Animal implements Serializable {
15. Dog() { System.out.print("d"); }
16. }
17. public class Beagle extends Dog { }
```
If an instance of class Beagle is created, then Serialized, then deSerialized, what is the result?

1) **ad**

2) **ada**

3) **add**

4) **adad**

5) **Compilation fails.**

6) **An exception is thrown at runtime.**

> **Your Selected options :: none** ✖
>
> **Correct Options   :: 2**

Click Here for Explanation

167) **Given:**
12. public class Wow {
13. public static void go(short n) {System.out.println("short");}
14. public static void go(Short n) {System.out.println("SHORT");}
15. public static void go(Long n) {System.out.println(" LONG");}
16. public static void main(String [] args) {
17. Short y = 6;
18. int z = 7;
19. go(y);
20. go(z);
21. }
22. }
What is the result?

　　1) **short LONG**

　　2) **SHORT LONG**

　　3) **Compilation fails.**

　　4) **An exception is thrown at runtime.**

　　　　Your Selected options　::　none　❌

　　　　Correct Options　　　::　3

Click Here for Explanation

168) **Given classes defined in two different files:**
1. package packageA;
2. public class Message {
3. String getText() { return "text"; }
4. }
and:
1. package packageB;
2. public class XMLMessage extends packageA.Message {
3. String getText() { return "<msg>text</msg>";}
4. public static void main(String[] args) {
5. System.out.println(new XMLMessage().getText());
6. }
7. }
What is the result of executing XMLMessage.main?

　　1) **text**

　　2) **An exception is thrown at runtime.**

　　3) **Compilation fails because of an error in line 2 of XMLMessage.**

　　4) **Compilation fails because of an error in line 3 of XMLMessage.**

　　　　Your Selected options　::　none　❌

　　　　Correct Options　　　::　4

Click Here for Explanation

169) **Click the Exhibit button.**
1. public class GoTest {
2. public static void main(String[] args) {
3. Sente a = new Sente(); a.go();
4. Goban b = new Goban(); b.go();
5. Stone c = new Stone(); c.go();
6. }
7. }
8.
9. class Sente implements Go {
10. public void go() { System.out.println(â€œgo in Sente.â€); }
11. }
12.
13. class Goban extends Sente {
14. public void go() { System.out.println(â€œgo in Gobanâ€); }
15. }
16.
17. class Stone extends Goban implements Go { }
18.
19. interface Go { public void go(); }
What is the result?

　　1) **go in Goban**
　　　go in Sente
　　　go in Sente

　　2) **go in Sente**

go in Sente
go in Goban

3) **go in Sente**
go in Goban
go in Goban

4) **go in Goban**
go in Goban
go in Sente

5) **Compilation fails because of an error in line 17.**

Your Selected options :: none ✖

Correct Options   :: 3

Click Here for Explanation

170) **Given:**
11. public class ItemTest {
12. private final int id;
13. public ItemTest(int id) { this.id = id; }
14. public void updateId(int newId) { id = newId; }
15.
16. public static void main(String[] args) {
17. ItemTest fa = new ItemTest(42);
18. fa.updateId(69);
19. System.out.println(fa.id);
20. }
21. }
What is the result?

1) **Compilation fails.**

2) **An exception is thrown at runtime.**

3) **The attribute id in the Item object remains unchanged.**

4) **The attribute id in the Item object is modified to the new value.**

5) **A new Item object is created with the preferred value in the id attribute.**

Your Selected options :: none ✖

Correct Options   :: 1

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**