DURGA ONLINE EXAMS

Test Your Knowledge

HOME

131) **Which three statements are true? (Choose three.)**

1) **A final method in class X can be abstract if and only if X is abstract.**
2) **A protected method in class X can be overridden by any subclass of X.**
3) **A private static method can be called only within other static methods in class X.**
4) **A non-static public final method in class X can be overridden in any subclass of X.**
5) **A public static method in class X can be called by a subclass of X without explicitly referencing the class X.**
6) **A method with the same signature as a private final method in class X can be implemented in a subclass of X.**
7) **A protected method in class X can be overridden by a subclass of A only if the subclass is in the same package as X.**

Your Selected options :: none

Correct Options :: 2, 5, 6

Click Here for Explanation

132) **Given:**

```
1.public class Blip {
2. protected int blipvert(int x) { return 0; }
3. }
4. class Vert extends Blip {
5. // insert code here
6. }
```

Which five methods, inserted independently at line 5, will compile? (Choose five.)

1) **public int blipvert(int x) { return 0; }**
2) **private int blipvert(int x) { return 0; }**
3) **private int blipvert(long x) { return 0; }**
4) **protected long blipvert(int x) { return 0; }**
5) **protected int blipvert(long x) { return 0; }**
6) **protected long blipvert(long x) { return 0; }**
7) **protected long blipvert(int x, int y) { return 0; }**

Your Selected options :: none

Correct Options :: 1, 3, 5, 6, 7

Click Here for Explanation

133) **Given:**

```
1. public class Base {
2. public static final String FOO = "foo";
3. public static void main(String[] args) {
4. Base b = new Base();
5. Sub s = new Sub();
6. System.out.print(Base.FOO);
7. System.out.print(Sub.FOO);
8. System.out.print(b.FOO);
9. System.out.print(s.FOO);
10. System.out.print(((Base)s).FOO);
11. } }
12. class Sub extends Base {public static final String FOO="bar";}
```

What is the result?

1) **foofoofoofoofoo**
2) **foobarfoobarbar**
3) **foobarfoofoofoo**
4) **foobarfoobarfoo**
5) **barbarbarbarbar**

6) **foofoofoobarbar**

7) **foofoofoobarfoo**

Your Selected options :: none

Correct Options :: 4

Click Here for Explanation

134) **Click the Exhibit button.**

```
1. public class Employee {
2. String name;
3. double baseSalary;
4. Employee(String name, double baseSalary) {
5. this.name = name;
6. this.baseSalary = baseSalary;
7. }
8. }
```

And:

```
1. public class Salesperson extends Employee {
2. double commission;
3. public Salesperson(String name, double baseSalary,
4. double commission) {
5. // insert code here
6. }
7. }
```

Which code, inserted at line 7, completes the Salesperson constructor?

1) **this.commission = commission;**

2) **superb();**
commission = commission;

3) **this.commission = commission;**
superb();

4) **super(name, baseSalary);**
this.commission = commission;

5) **super();**
this.commission = commission;

6) **this.commission = commission;**
super(name, baseSalary);

Your Selected options :: none

Correct Options :: 4

Click Here for Explanation

135) **Given:**

```
1. class TestA {
2. public void start() { System.out.println("TestA"); }
3. }
4. public class TestB extends TestA {
5. public void start() { System.out.println("TestB"); }
6. public static void main(String[] args) {
7. ((TestA)new TestB()).start();
8. }
9. }
```

What is the result?

1) **TestA**

2) **TestB**

3) **Compilation fails.**

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

Your Selected options :: none

Correct Options :: 2

Click Here for Explanation

136) **Click the Exhibit button.**
Which three code fragments, added individually at line 29, produce the output 100? (Choose three.)

```
10. class Inner {
11.   private int x;
12.   public void setX( int x ) { this.x = x;
}
13.   public int getX() { return x; }
14. }
15.
16. class Outer {
17.   private Inner y;
18.   public void setY( Inner y ) { this.y =
y; }
19.   public Inner getY() { return y; }
20. }
21.
22. public class Gamma {
23.   public static void main( String[] args
) {
24.     Outer o = new Outer();
25.     Inner i = new Inner();
26.     int n = 10;
27.     i.setX( n );
28.     o.setY( i );
29.     // insert code here
30.   System.out.println( o.getY().getX() );
31.   }
32. }
```

1) **n = 100;**

2) **i.setX(100);**

3) **o.getY().setX(100);**

4) **i = new Inner(); i.setX(100);**

5) **o.setY(i); i = new Inner(); i.setX(100);**

6) **i = new Inner(); i.setX(100); o.setY(i);**

Your Selected options :: none

Correct Options :: 2, 3, 6

Click Here for Explanation

137) **Given:**

```
5. class Atom {
6. Atom() { System.out.print("atom "); }
7. }
8. class Rock extends Atom {
9. Rock(String type) { System.out.print(type); }
10. }
11. public class Mountain extends Rock {
12. Mountain() {
13. super("granite ");
14. new Rock("granite ");
15. }
16. public static void main(String[] a) { new Mountain(); }
17. }
```

What is the result?

1) **Compilation fails.**

2) **atom granite**

3) **granite granite**

4) **atom granite granite**

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

6) **atom granite atom granite**

Your Selected options :: none

Correct Options :: 6

Click Here for Explanation

138) **Given:**

```
31. class Foo {
32. public int a = 3;
33. public void addFive() { a += 5; System.out.print("f "); }
34. }
35. class Bar extends Foo {
36. public int a = 8;
37. public void addFive() { this.a += 5; System.out.print("b " ); }
38. }
Invoked with:
Foo f = new Bar();
f.addFive();
System.out.println(f.a);
```

What is the result?

1) **b 3**
2) **b 8**
3) **b 13**
4) **f 3**
5) **f 8**
6) **f 13**
7) **Compilation fails.**
8) **An exception is thrown at runtime.**

Your Selected options :: none
Correct Options :: 1

Click Here for Explanation

139) **A company that makes Computer Assisted Design (CAD) software has, within its application, some utility classes that are used to perform 3D rendering tasks. The company's chief scientist has just improved the performance of one of the utility classes' key rendering algorithms, and has assigned a programmer to replace the old algorithm with the new algorithm. When the programmer begins researching the utility classes, she is happy to discover that the algorithm to be replaced exists in only one class. The programmer reviews that class's API, and replaces the old algorithm with the new algorithm, being careful that her changes adhere strictly to the class's API. Once testing has begun, the programmer discovers that other classes that use the class she changed are no longer working properly. What design flaw is most likely the cause of these new bugs?**

1) **Inheritance**
2) **Tight coupling**
3) **Low cohesion**
4) **High cohesion**
5) **Loose coupling**
6) **Object immutability**

Your Selected options :: none
Correct Options :: 2

Click Here for Explanation

140) **Click the Exhibit button.**

```
11. class Payload {
12. private int weight;
13. public Payload(int wt) { weight = wt; }
14. public void setWeight(mt w) { weight = w; }
15. public String toString { return Integer.toString(weight); }
16. }
17.
18. public class TestPayload {
19. static void changePayload(Payload p) {
20. /* insert code here */
21. }
22.
23. public static void main(String[] args) {
24. Payload p = new Payload();
25. p.setWeight(1024);
26. changePayload(p);
27. System.out.println(â€The value of p is â€œ+ p);
28. }
29. }
```

Which statement, placed at line 20, causes the code to print â€œThe value of p is 420.â€?

1) **p.setWeight(420);**
2) **p.changePayload(420);**
3) **p = new Payload(420);**
4) **Payload.setWeight(420);**
5) **p = Payload.setWeight(420);**
6) **p = new Payload();**
   p.setWeight(420);

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**