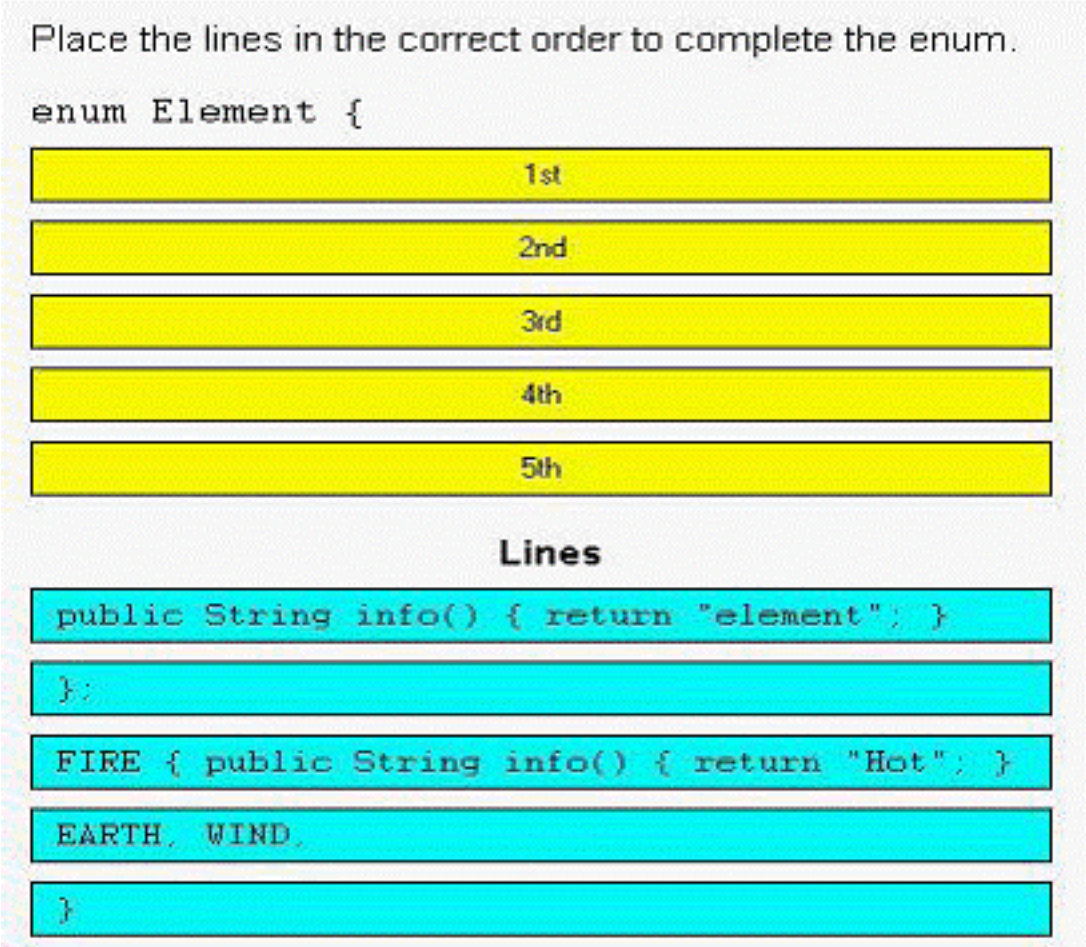
**For more details on SUN Certifications, visit JavaScjpDumps**

**Q: 1 Click the Task button.**




**Solution:**

```
enum Element{
        EARTH,WIND,
        FIRE{public String info(){return "Hot";}
        };
        public String info(){return "element";}
}
```

**Q: 2 Given:**

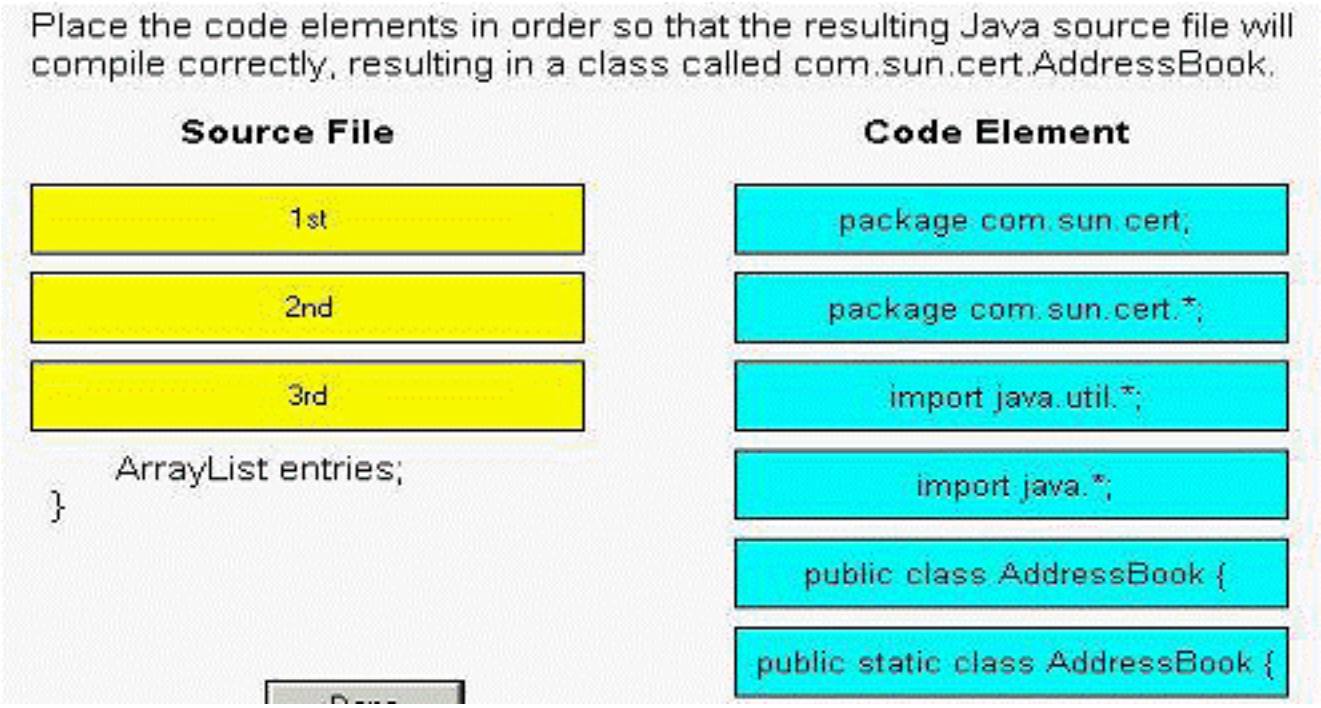
```
10. package com.sun.scjp;
11. public class Geodetics {
12. public static final double DIAMETER = 12756.32; // kilometers
13. }
```

**Which two correctly access the DIAMETER member of the Geodetics class? (Choose two.)**

A. import com.sun.scjp.Geodetics;

public class TerraCarta {
public double halfway()
{ return Geodetics.DIAMETER/2.0; }
B. import static com.sun.scjp.Geodetics;
public class TerraCarta{
public double halfway() { return DIAMETER/2.0; } }
C. import static com.sun.scjp.Geodetics.*;
public class TerraCarta {
public double halfway() { return DIAMETER/2.0; } }
D. package com.sun.scjp;
public class TerraCarta {
public double halfway() { return DIAMETER/2.0; } }
**Answer: A, C**

**Q: 3 Click the Task button.**




**Solution:**
**package com.sun.cert;**
**import java.util.*;**
**public class AddressBook{**
**ArrayList entries;**
**}**

**Q: 4 Which two classes correctly implement both the java.lang.Runnable and the java.lang.Clonable interfaces? (Choose**
A. public class Session

implements Runnable, Clonable {
public void run();
public Object clone();
}
B. public class Session
extends Runnable, Clonable {
public void run() { /* do something */ }
public Object clone() { /* make a copy */ }
C. public class Session
implements Runnable, Clonable {
public void run() { /* do something */ }
public Object clone() { /* make a copy */ }
D. public abstract class Session
implements Runnable, Clonable {
public void run() { /* do something */ }
public Object clone() { /*make a copy */ }
E. public class Session
implements Runnable, implements Clonable {
public void run() { /* do something */ }
public Object clone() { /* make a copy */ }
**Answer: C, D**

**Q: 5 Given classes defined in two different files:**
**1. package util;**
**2. public class BitUtils {**
**3. private static void process(byte[] b) {}**
**4. }**
**1. package app;**
**2. public class SomeApp {**
**3. public static void main(String[] args) {**
**4. byte[] bytes = new byte[256];**
**5. // insert code here**
**6. }**
**7. }**
**What is required at line 5 in class SomeApp to use the process method of BitUtils?**
A. process(bytes);
B. BitUtils.process(bytes);
C. app.BitUtils.process(bytes);
D. util.BitUtils.process(bytes);
E. import util.BitUtils.*; process(bytes);
F. SomeApp cannot use the process method in BitUtils.
**Answer: F**

**Q: 6 Given:**

```
11. class Cup { }
12. class PoisonCup extends Cup { }
...
21. public void takeCup(Cup c) {
22. if (c instanceof PoisonCup) {
23. System.out.println("Inconceivable!");
24. } else if (c instanceof Cup) {
25. System.out.println("Dizzying intellect!");
26. } else {
27. System.exit(0);
28. }
29. }
```

**And the execution of the statements:**

```
Cup cup = new PoisonCup();
takeCup(cup);
```

**What is the output?**

A. Inconceivable!
B. Dizzying intellect!
C. The code runs with no output.
D. An exception is thrown at runtime.
E. Compilation fails because of an error in line 22.

**Answer: A**

**Q: 7 Click the Exhibit button.**

```
public class A
{
        private int counter=0;
        public static int getInstanceCount()
        {
        return counter;
         }
         public A()
         {
         counter++;
        }
 }
```

**Given this code from Class B:**

```
25. A a1 = new A();
26. A a2 = new A();
27. A a3 = new A();
28. System.out.println(A.getInstanceCount());
```

**What is the result?**

A. Compilation of class A fails.
B. Line 28 prints the value 3 to System.out.
C. Line 28 prints the value 1 to System.out.
D. A runtime error occurs when line 25 executes.
E. Compilation fails because of an error on line 28.
**Answer: A**

**Q:8 Given:**
```
11. String[] elements = { "for", "tea", "too" };
12. String first = (elements.length > 0) ? elements[0] : null;
```
**What is the result?**
A. Compilation fails.
B. An exception is thrown at runtime.
C. The variable first is set to null.
D. The variable first is set to elements[0].
**Answer: D**

**Q:09 Given:**
```
11. interface DeclareStuff {
12. public static final int EASY = 3;
13. void doStuff(int t); }
14. public class TestDeclare implements DeclareStuff {
15. public static void main(String [] args) {
16. int x = 5;
17. new TestDeclare().doStuff(++x);
18. }
19. void doStuff(int s) {
20. s += EASY + ++s;
21. System.out.println("s " + s);
22. }
23. }
```
**What is the result?**
A. s 14
B. s 16
C. s 10
D. Compilation fails.
E. An exception is thrown at runtime.
**Answer: D**

**Q: 10 Given:**
```
1. public class TestString1 {
2. public static void main(String[] args) {
3. String str = "420";
```

**4. str += 42;**
**5. System.out.print(str);**
**6. }**
**7. }**
**What is the output?**
A. 42
B. 420
C. 462
D. 42042
E. Compilation fails.
F. An exception is thrown at runtime.
**Answer: D**

**Q: 11 Given:**
**11. class Converter {**
**12. public static void main(String[] args) {**
**13. Integer i = args[0];**
**14. int j = 12;**
**15. System.out.println("It is " + (j==i) + " that j==i.");**
**16. }**
**17. }**
**What is the result when the programmer attempts to compile the code and run it with the command java Converter 12?**
A. It is true that j==i.
B. It is false that j==i.
C. An exception is thrown at runtime.
D. Compilation fails because of an error in line 13.
**Answer: D**

**Q: 12 Given:**
**10. int x = 0;**
**11. int y = 10;**
**12. do {**
**13. y--;**
**14. ++x;**
**15. } while (x < 5);**
**16. System.out.print(x + "," + y);**
**What is the result?**
A. 5,6
B. 5,5
C. 6,5
D. 6,6
**Answer: B**

**Q: 13 Given:**

```
1. public interface A {
2. String DEFAULT_GREETING = "Hello World";
3. public void method1();
4. }
```

**A programmer wants to create an interface called B that has A as its parent. Which interface declaration is correct?**

A. public interface B extends A {}
B. public interface B implements A {}
C. public interface B instanceOf A {}
D. public interface B inheritsFrom A {}

**Answer: A**

**Q: 14 Given:**

```
11. public enum Title {
12. MR("Mr."), MRS("Mrs."), MS("Ms.");
13. private final String title;
14. private Title(String t) { title = t; }
15. public String format(String last, String first) {
16. return title + " " + first + " " + last;
17. }
18. }
19. public static void main(String[] args) {
20. System.out.println(Title.MR.format("Doe", "John"));
21. }
```

**What is the result?**

A. Mr. John Doe
B. An exception is thrown at runtime.
C. Compilation fails because of an error in line 12.
D. Compilation fails because of an error in line 15.
E. Compilation fails because of an error in line 20.

**Answer: A**

**Q: 15 Given:**

```
1. package test;
2.
3. class Target {
4. public String name = "hello";
5. }
```

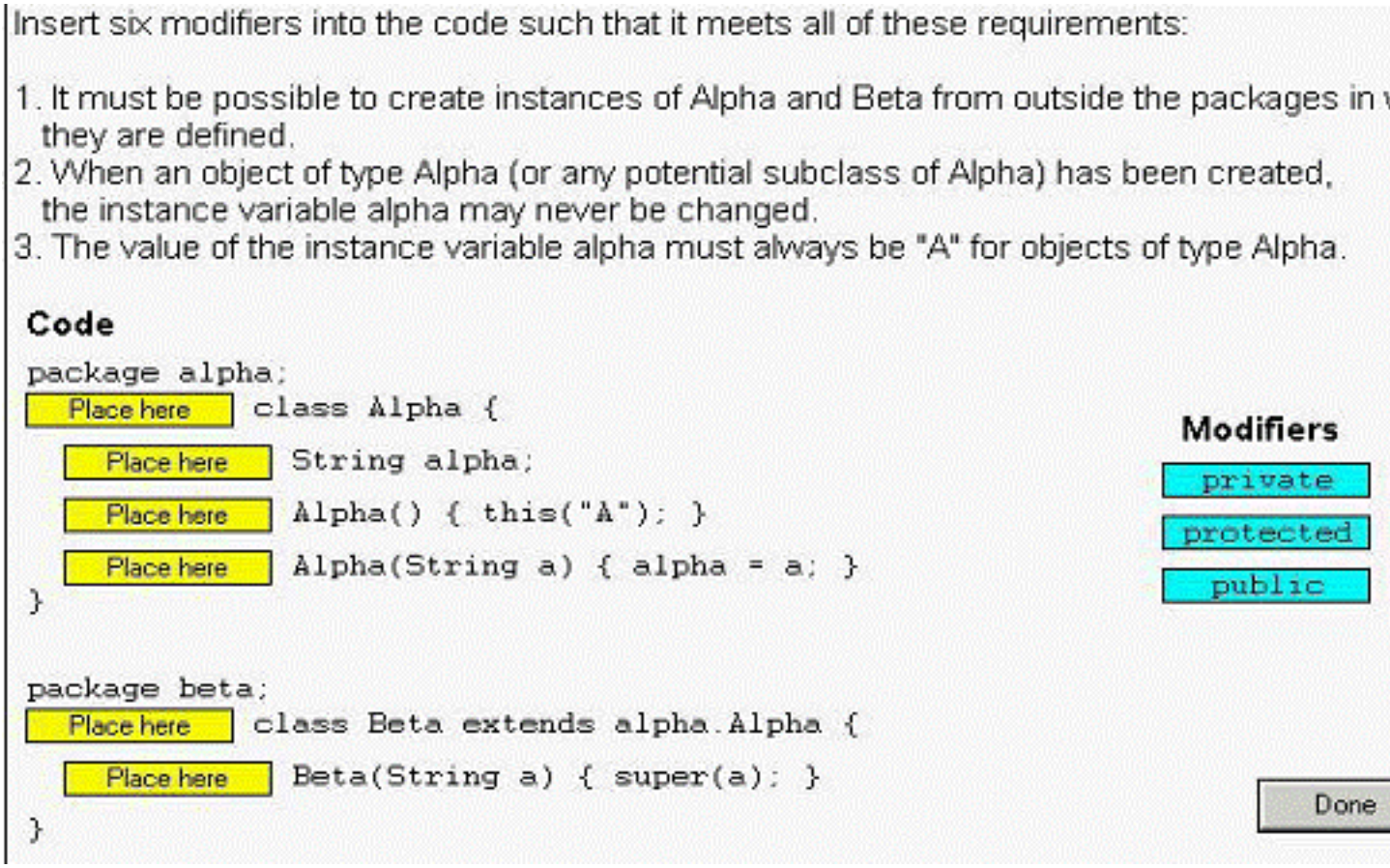
**What can directly access and change the value of the variable name?**

A. any class
B. only the Target class
C. any class in the test package
D. any class that extends Target

**Answer: C**

**Q: 16 Given:**
**11. public class Ball{**
**12. public enum Color { RED, GREEN, BLUE };**
**13. public void foo(){**
**14. // insert code here**
**15. { System.out.println(c); }**
**16. }**
**17. }**
**Which code inserted at line 14 causes the foo method to print RED, GREEN, and BLUE?**
A. for( Color c : Color.values() )
B. for( Color c = RED; c <= BLUE; c++ )
C. for( Color c ; c.hasNext() ; c.next() )
D. for( Color c = Color[0]; c <= Color[2]; c++ )
E. for( Color c = Color.RED; c <= Color.BLUE; c++ )
**Answer: A**

**Q: 17 Click the Task button.**



**Solution:**

```
package alpha;
public class Alpha{
private String alpha;
public  Alpha( ){ this("A") ; }
protected Alpha(String a){ alpha=a;  }
}
package beta;
public class Beta extends alpha.Alpha{
private Beta(String a){ super(a);  }
}
```

**Q: 18 Given:**

```
1. public class Target {
2. private int i = 0;
3. public int addOne(){
4. return ++i;
5. }
6. }
```

**And:**

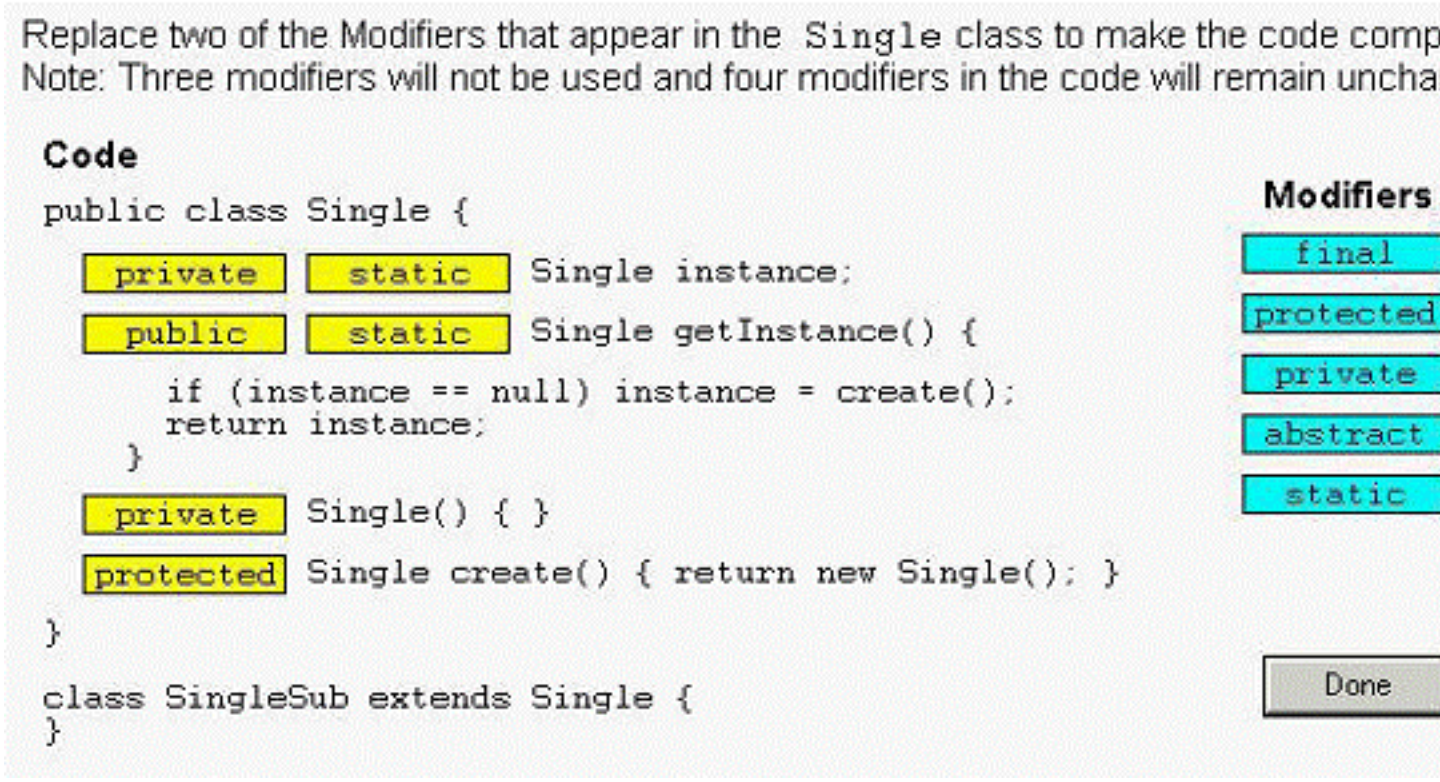
```
1. public class Client {
2. public static void main(String[] args){
3. System.out.println(new Target().addOne());
4. }
5. }
```

**Which change can you make to Target without affecting Client?**

A. Line 4 of class Target can be changed to return i++;
B. Line 2 of class Target can be changed to private int i = 1;
C. Line 3 of class Target can be changed to private int addOne(){
D. Line 2 of class Target can be changed to private Integer i = 0;

**Answer: D**

**Q: 19 Click the Task button.**




**Solution:**

```
public class Single{
        private static Single instance;
        public static Single getInstance( ){
 if(instance==null) instance = create( );
        return instance;
  }
protectedSingle( ) { }
staticSingle create ( ) { return new Single ( ) ; }
}
class SingleSub extends Shape{
 }
```

**Q: 20 Given:**

```
12. public class Test {
13. public enum Dogs {collie, harrier};
14. public static void main(String [] args) {
15. Dogs myDog = Dogs.collie;
16. switch (myDog) {
17. case collie:
18. System.out.print("collie ");
19. case harrier:
20. System.out.print("harrier ");
```

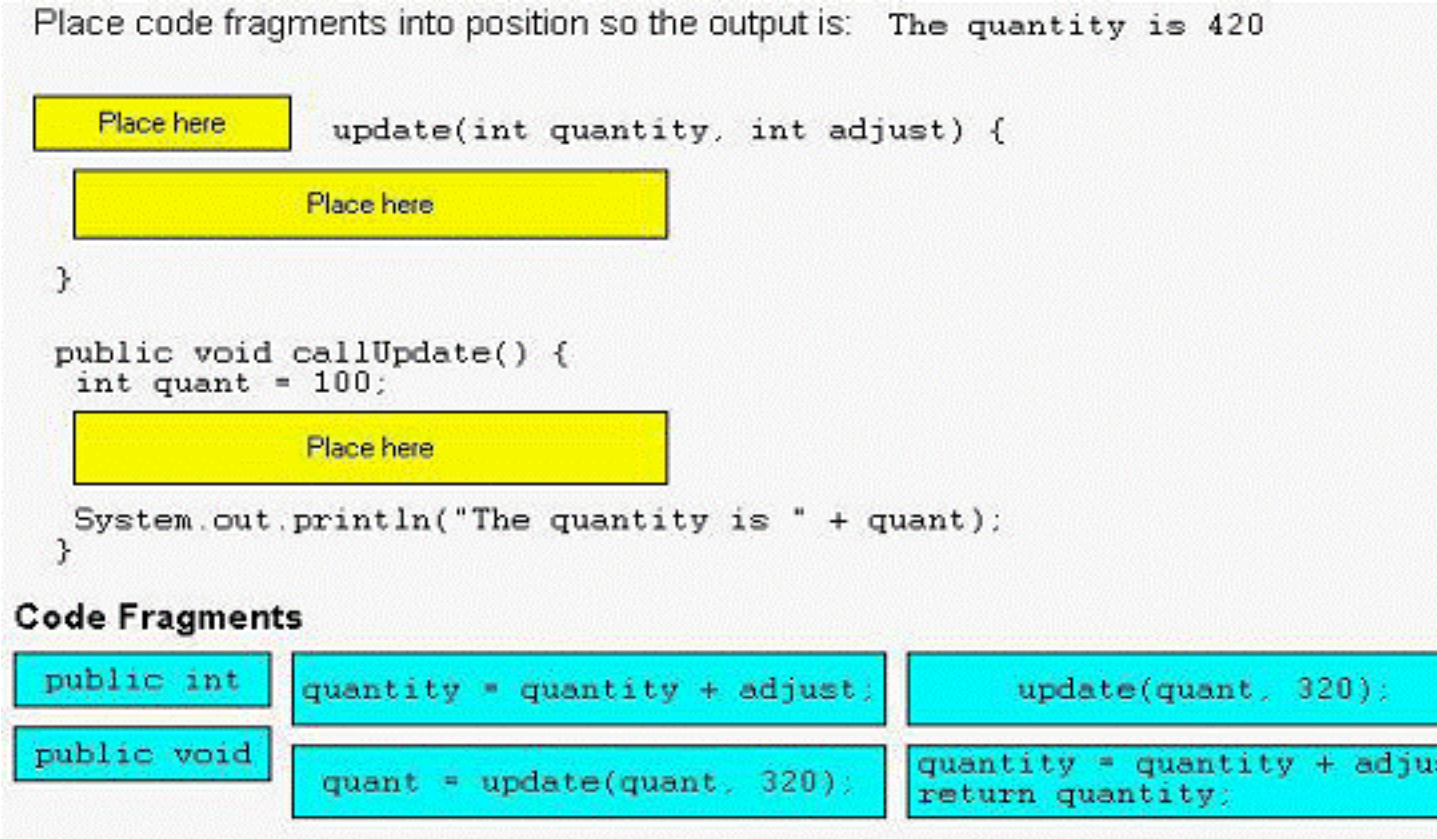
**21. }**
**22. }**
**23. }**
**What is the result?**
A. collie B. harrier
C. Compilation fails. D. collie harrier
E. An exception is thrown at runtime.
**Answer: D**

**Q: 21 Click the Exhibit button.**
**Given:**
**ClassA a = new ClassA();**
**a.methodA();**
**What is the result?**

```
10. public class ClassA {
11.   public void methodA() {
12.     ClassB classB = new ClassB();
13.     classB.getValue();
14.   }
15. }

And:

20. class ClassB {
21.   public ClassC classC;
22.
23.   public String getValue() {
24.     return classC.getValue();
25.   }
26. }

And:

30. class ClassC {
31.   public String value;
32.
33.   public String getValue() {
34.     value = "ClassB";
35.     return value;
36.   }
37. }
```

A. Compilation fails.
B. ClassC is displayed.
C. The code runs with no output.
D. An exception is thrown at runtime.
**Answer: D**

**Q: 22 Click the Task button.**




**Solution:**

```
public int update(int quantity,int adjust){
quantity=quantity+adjust;
return quantity;
}
public void call Update( ) {
int quant=100;
quant=update(quant,320);
System.out.println("the quantity is " +quant);
}
```

**Q: 23 Given:**

```
1. package sun.scjp;
2. public enum Color { RED, GREEN, BLUE }
1. package sun.beta;
2. // insert code here
3. public class Beta {
4. Color g = GREEN;
5. public static void main( String[] argv)
6. { System.out.println( GREEN); }
```

**7. }**
**The class Beta and the enum Color are in different packages.**
**Which two code fragments, inserted individually at line 2 of the Beta declaration, will allow this code to compile? (Choose two.)**
A. import sun.scjp.Color.*;
B. import static sun.scjp.Color.*;
C. import sun.scjp.Color; import static sun.scjp.Color.*;
D. import sun.scjp.*; import static sun.scjp.Color.*;
E. import sun.scjp.Color; import static sun.scjp.Color.GREEN;
**Answer: CE**

**Question 24**
**Given:**
**10. public class Fabric**
**11. public enum Color {**
**12. RED(0xff0000), GREEN(0x00ff00), BLUE(0x0000ff);**
**13. private final int rgb;**
**14. Color( int rgb) { this.rgb = rgb; }**
**15. public int getRGB() { return rgb; }**
**16. };**
**17. public static void main( String[] argv) {**
**18. // insert code here**
**19. }**
**20. }**
**Which two code fragments, inserted independently at line 18, allow the Fabric class to compile? (Choose two.)**
A. Color skyColor = BLUE;
B. Color treeColor = Color.GREEN;
C. Color purple = new Color( 0xff00ff);
D. if( RED.getRGB() < BLUE.getRGB() ) {}
E. Color purple = Color.BLUE + Color.RED;
F. if( Color.RED.ordinal() < Color.BLUE.ordinal() ) {}
**Answer: BF**

**25. Given the following,**
**1. interface Base {**
**2. boolean m1 ();**
**3. byte m2(short s);**
**4. }**
**Which code fragments will compile? (Choose all that apply.)**
A. interface Base2 implements Base { }
B. abstract class Class2 extends Base {
public boolean m1() { return true; } }

C. abstract class Class2 implements Base { }
D. abstract class Class2 implements Base {
public boolean m1() { return (true); } }
E. class Class2 implements Base {
boolean m1() { return false; }
byte m2(short s) { return 42; } }
**Answer:**
-> **C** and **D** are correct. **C** is correct because an abstract class doesn't have to implement any or all of its interface's methods. **D** is correct because the method is correctly implemented.
-> **A** is incorrect because interfaces don't implement anything. **B** is incorrect because classes don't extend interfaces. **E** is incorrect because interface methods are implicitly public, so the methods being implemented must be public.

**26. Which declare a compilable abstract class? (Choose all that apply.)**
A. public abstract class Canine { public Bark speak(); }
B. public abstract class Canine { public Bark speak() { } }
C. public class Canine { public abstract Bark speak(); }
D. public class Canine abstract { public abstract Bark speak(); }
**Answer:**
->**B** is correct. abstract classes don't have to have any abstract methods.
-> **A** is incorrect because abstract methods must be marked as such. **C** is incorrect becauseyou can't have an abstract method unless the class is abstract. **D** is incorrect because thekeyword abstract must come before the classname. (Objective 1.1)

**27. Which is true? (Choose all that apply.)**
A. "X extends Y" is correct if and only if X is a class and Y is an interface.
B. "X extends Y" is correct if and only if X is an interface and Y is a class.
C. "X extends Y" is correct if X and Y are either both classes or both interfaces.
D. "X extends Y" is correct for all combinations of X and Y being classes and/or interfaces.
**Answer:**
-> **C** is correct.
-> **A** is incorrect because classes implement interfaces, they don't extend them.
**B** is incorrect because interfaces only "inherit from" other interfaces.
**D** is incorrect based on the preceding rules.

**28. Given:**
**1. enum Animals {**
**2. DOG("woof"), CAT("meow"), FISH("burble");**
**3. String sound;**
**4. Animals(String s) { sound = s; }**
**5. }**
**6. class TestEnum {**
**7. static Animals a;**

**8. public static void main(String [] args) {**
**9. System.out.println(a.DOG.sound + " " + a.FISH.sound);**
**10. }**
**11. }**
**What is the result?**
A. woof burble
B. Multiple compilation errors
C. Compilation fails due to an error on line 2
D. Compilation fails due to an error on line 3
E. Compilation fails due to an error on line 4
F. Compilation fails due to an error on line 9
**Answer:**
-> **A** is correct; enums can have constructors and variables.
->**B, C, D, E,** and **F** are incorrect; these lines all use correct syntax.

**29. Given:**
**1. enum A { A }**
**2. class E2 {**
**3. enum B { B }**
**4. void C() {**
**5. enum D { D }**
**6. }**
**7. }**
**Which statements are true? (Choose all that apply.)**
A. The code compiles.
B. If only line 1 is removed the code compiles.
C. If only line 3 is removed the code compiles.
D. If only line 5 is removed the code compiles.
E. If lines 1 and 3 are removed the code compiles.
F. If lines 1, 3 and 5 are removed the code compiles.
**Answer**
->**D** and **F** are correct. Line 5 is the only line that will not compile, because enums cannot belocal to a method.
->**A, B, C** and **E** are incorrect based on the above.