Exam Code: 310-056

Exam Name: Sun Certified Programmer for J2SE 5.0 - Upgrade Exam

Vendor: Sun

Version: DEMO

Part: A

1: Given:

```
1.  interface A { public void aMethod(); }
2.  interface B { public void bMethod(); }
3.  interface C extends A,B { public void cMethod(); }
4.  class D implements B {
5.      public void bMethod(){}
6.  }
7.  class E extends D implements C {
8.      public void aMethod(){}
9.      public void bMethod(){}
10.       public void cMethod(){}
11.   }
```

What is the result?

A.Compilation fails because of an error in line 3.

B.Compilation fails because of an error in line 7.

C.Compilation fails because of an error in line 9.

D.If you define D e = new E(), then e.bMethod() invokes the version of bMethod() defined in Line 5.

E.If you define D e = (D)(new E()), then e.bMethod() invokes the version of bMethod() defined in Line 5.

F.If you define D e = (D)(new E()), then e.bMethod() invokes the version of bMethod() defined in Line 9.

Correct Answers: F

2: Given:

```
20. public class CreditCard {
21.
22.     private String cardID;
23.     private Integer limit;
24.     public String ownerName;
25.
26.     public void setCardInformation(String cardID,
27.                                    String ownerName,
28.                                    Integer limit) {
29.         this.cardID = cardID;
30.         this.ownerName = ownerName;
31.         this.limit = limit;
32.     }
33. }
```

Which statement is true?

A.The class is fully encapsulated.

B.The code demonstrates polymorphism.

C.The ownerName variable breaks encapsulation.

D.The cardID and limit variables break polymorphism.

E.The setCardInformation method breaks encapsulation.

Correct Answers: C

3: Given:

```
1. class Super {
2.     private int a;
3.     protected Super(int a) { this.a = a; }
4. }
  ...
11. class Sub extends Super {
12.     public Sub(int a) { super(a); }
13.     public Sub() { this.a = 5; }
14. }
```

Which two, independently, will allow Sub to compile? (Choose two.)

A.Change line 2 to:
public int a;

B.Change line 2 to:
protected int a;

C.Change line 13 to:
public Sub() { this(5); }

D.Change line 13 to:
public Sub() { super(5); }

E.Change line 13 to:
public Sub() { super(a); }

Correct Answers: C D

4: Which two statements are true? (Choose two.)

A.An encapsulated, public class promotes re-use.

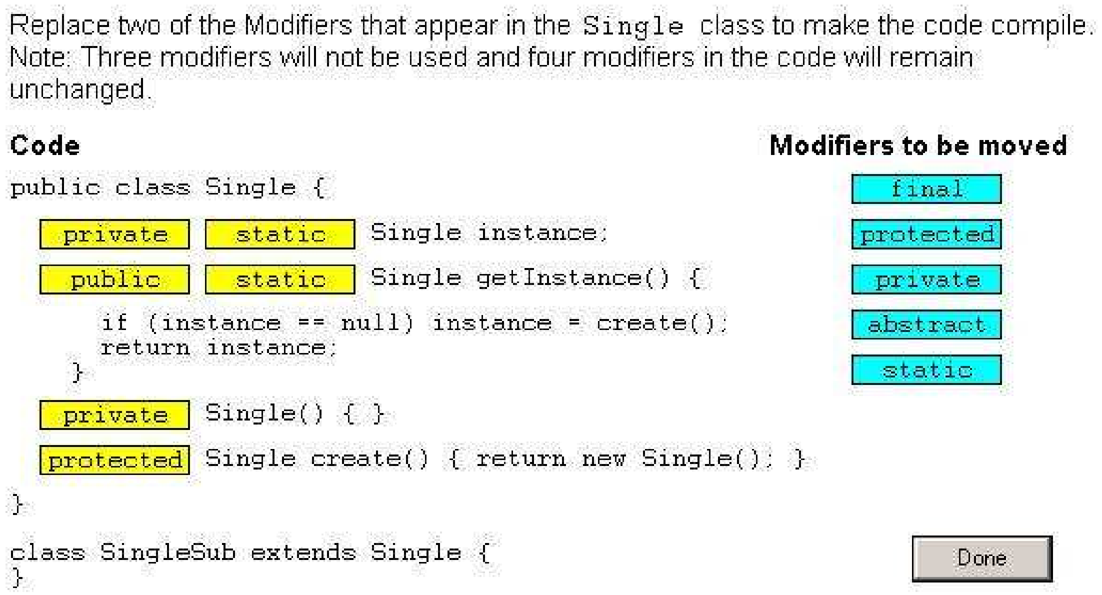
B.Classes that share the same interface are always tightly encapsulated.

C.An encapsulated class allows subclasses to overload methods, but does NOT allow overriding methods.

D.An encapsulated class allows a programmer to change an implementation without affecting outside code.

Correct Answers: A D

5: Click the Task button.



Replace two of the Modifiers that appear in the `Single` class to make the code compile. Note: Three modifiers will not be used and four modifiers in the code will remain unchanged.

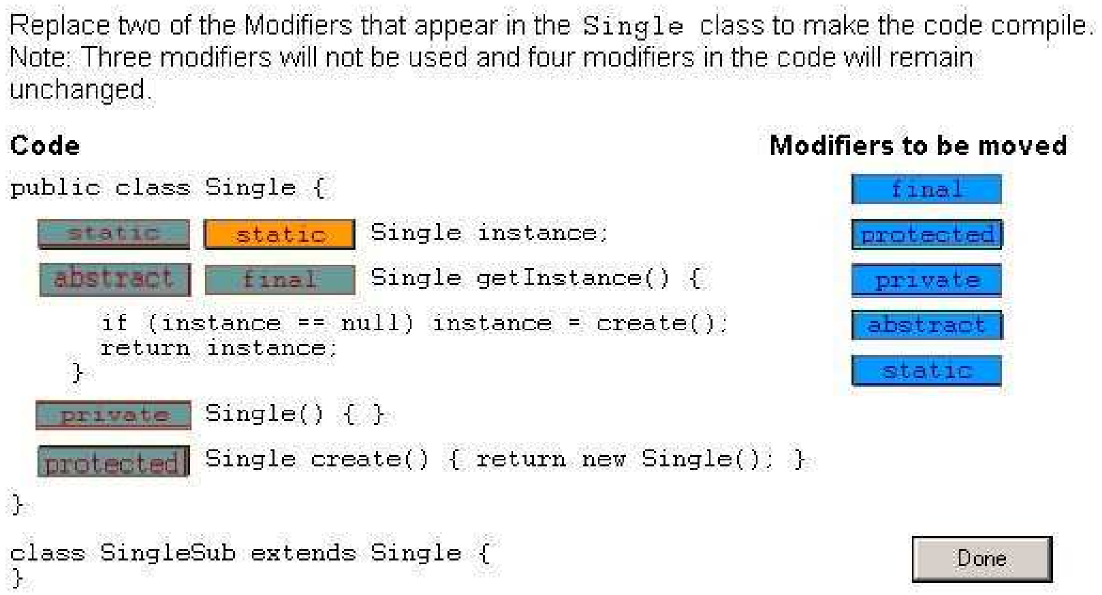
Code

```
public class Single {
   private  static  Single instance;
   public   static  Single getInstance() {
      if (instance == null) instance = create();
      return instance;
   }
   private  Single() { }
   protected  Single create() { return new Single(); }
}

class SingleSub extends Single {
}
```

Modifiers to be moved

Correct Answers:



Replace two of the Modifiers that appear in the `Single` class to make the code compile. Note: Three modifiers will not be used and four modifiers in the code will remain unchanged.

Code

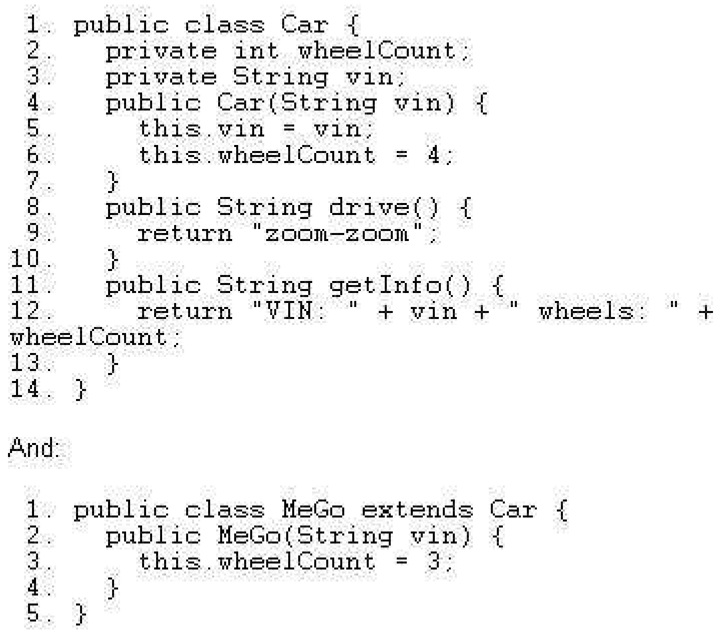
```
public class Single {
   static   static  Single instance;
   abstract final   Single getInstance() {
      if (instance == null) instance = create();
      return instance;
   }
   private  Single() { }
   protected  Single create() { return new Single(); }
}

class SingleSub extends Single {
}
```

Modifiers to be moved

6: Click the Exhibit button.

What two must the programmer do to correct the compilation errors? (Choose two.)



```
 1. public class Car {
 2.   private int wheelCount;
 3.   private String vin;
 4.   public Car(String vin) {
 5.     this.vin = vin;
 6.     this.wheelCount = 4;
 7.   }
 8.   public String drive() {
 9.     return "zoom-zoom";
10.   }
11.   public String getInfo() {
12.     return "VIN: " + vin + " wheels: " +
wheelCount;
13.   }
14. }
```

And:

```
 1. public class MeGo extends Car {
 2.   public MeGo(String vin) {
 3.     this.wheelCount = 3;
 4.   }
 5. }
```

A.insert a call to this() in the Car constructor

B.insert a call to this() in the MeGo constructor

C.insert a call to super() in the MeGo constructor

D.insert a call to super(vin) in the MeGo constructor

E.change the wheelCount variable in Car to protected

F.change line 3 in the MeGo class to super.wheelCount = 3;

Correct Answers: D E

7: Given:

```
1. class ClassA {
2.     public int numberOfInstances;
3.     protected ClassA(int numberOfInstances) {
4.         this.numberOfInstances = numberOfInstances;
5.     }
6. }
7. public class ExtendedA extends ClassA {
8.     private ExtendedA(int numberOfInstances) {
9.         super(numberOfInstances);
10.     }
11.     public static void main(String[] args) {
12.         ExtendedA ext = new ExtendedA(420);
13.         System.out.print(ext.numberOfInstances);
14.     }
15. }
```

Which statement is true?

A.420 is the output.

B.An exception is thrown at runtime.

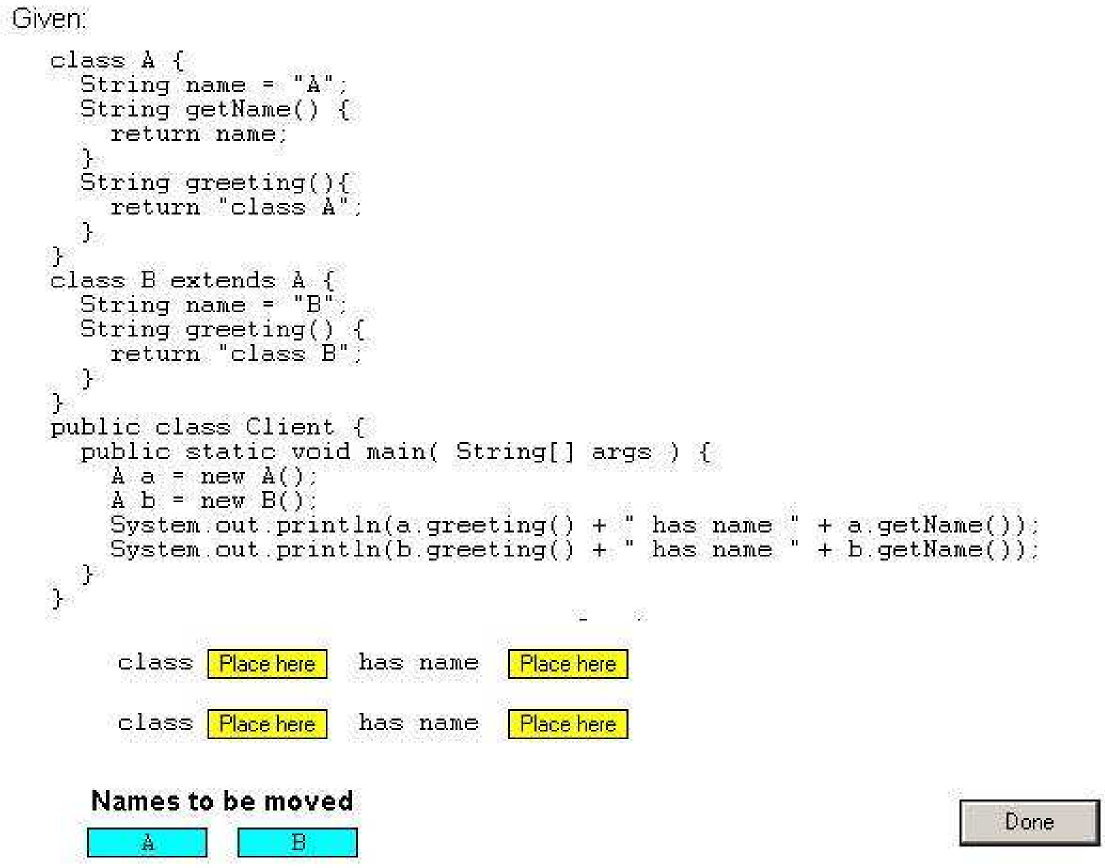
C.All constructors must be declared public.

D.Constructors CANNOT use the private modifier.

E.Constructors CANNOT use the protected modifier.

Correct Answers: A

8: Click the Task button.

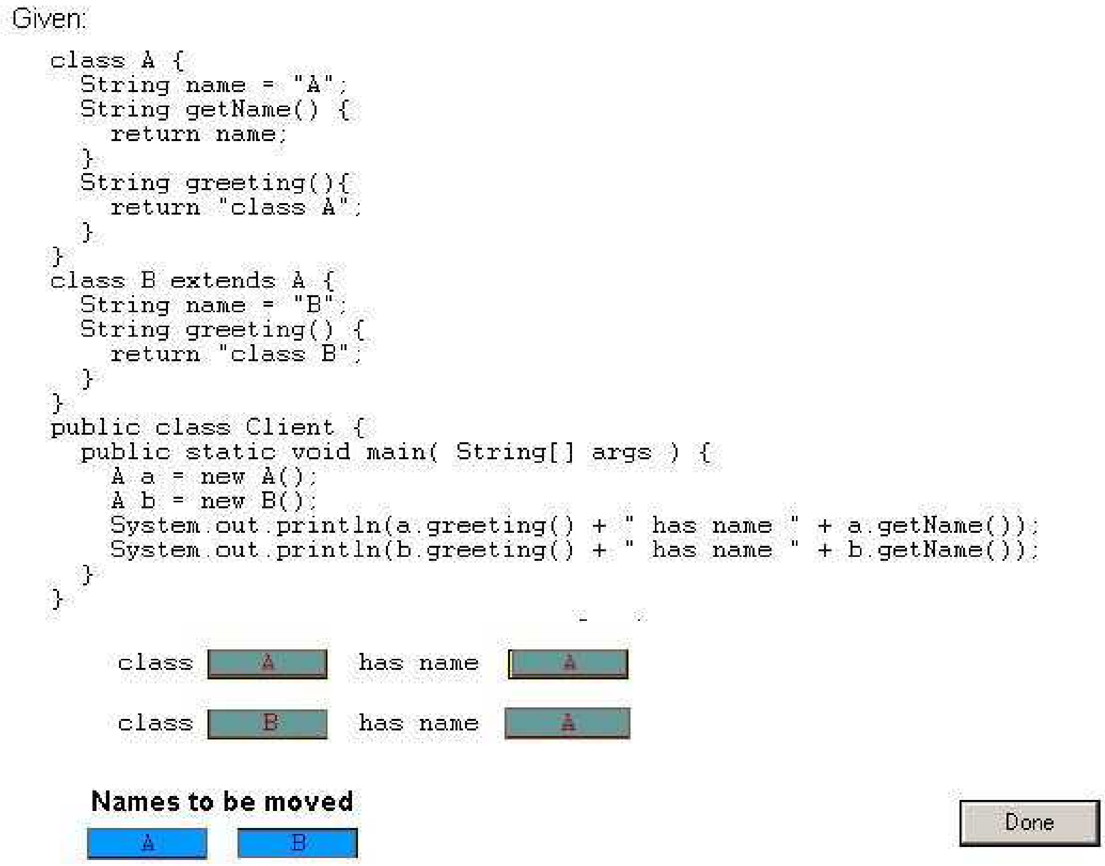


Given:

```
class A {
  String name = "A";
  String getName() {
    return name;
  }
  String greeting(){
    return "class A";
  }
}
class B extends A {
  String name = "B";
  String greeting() {
    return "class B";
  }
}
public class Client {
  public static void main( String[] args ) {
    A a = new A();
    A b = new B();
    System.out.println(a.greeting() + " has name " + a.getName());
    System.out.println(b.greeting() + " has name " + b.getName());
  }
}
```

class Place here has name Place here
class Place here has name Place here
Names to be moved
A
B
Done

Correct Answers:



Given:

```
class A {
  String name = "A";
  String getName() {
    return name;
  }
  String greeting(){
    return "class A";
  }
}
class B extends A {
  String name = "B";
  String greeting() {
    return "class B";
  }
}
public class Client {
  public static void main( String[] args ) {
    A a = new A();
    A b = new B();
    System.out.println(a.greeting() + " has name " + a.getName());
    System.out.println(b.greeting() + " has name " + b.getName());
  }
}
```

```
class A has name A
class B has name A
```

9: Given a valid DateFormat object named df, and

16. Date d = new Date(0L);

17. String ds = "December 15, 2004";

18. // insert code here

What updates d's value with the date represented by ds?

A.18. d = df.parse(ds);

B.18. d = df.getDate(ds);

C.18. try {

19.   d = df.parse(ds);

20. }   catch(ParseException e) {   };

D.18. try {

19.   d = df.getDate(ds);

20. }   catch(ParseException e) {   };

Correct Answers: C

10: Which three statements concerning the use of the java.io.Serializable interface are true? (Choose three.)

A.Objects from classes that use aggregation cannot be serialized.

B.An object serialized on one JVM can be successfully deserialized on a different JVM.

C.The values in fields with the volatile modifier will NOT survive serialization and deserialization.

D.The values in fields with the transient modifier will NOT survive serialization and deserialization.

E.It is legal to serialize an object of a type that has a supertype that does NOT implement java.io.Serializable.

Correct Answers: B D E