D. The interface used to implement sorting allows this class to define many different sort sequences.

Answer: A,C

QUESTION: 235

Given:

```
import java.util.*;
public class Explorer3 {
public static void main(String[] args) {
TreeSet<Integer> s = new TreeSet<Integer>();
TreeSet<Integer> subs = new TreeSet<Integer>();
for(int i = 606; i < 613; i++)
if(i%2 == 0) s.add(i);
subs = (TreeSet)s.subSet(608, true, 611, true);
subs.add(629);
System.out.println(s + " " + subs);
}
}
```

What is the result?

A. Compilation fails.
B. An exception is thrown at runtime.
C. [608, 610, 612, 629] [608, 610]
D. [608, 610, 612, 629] [608, 610, 629]
E. [606, 608, 610, 612, 629] [608, 610]
F. [606, 608, 610, 612, 629] [608, 610, 629]

Answer: B

QUESTION: 236

Given:

```
import java.util.*;

public class LetterASort{
public static void main(String[] args) {
ArrayList<String> strings = new ArrayList<String>();
strings.add("aAaA");
strings.add("AaA");
strings.add("aAa");
strings.add("AAaa");
Collections.sort(strings);
for (String s : strings) { System.out.print(s + " "); }
12. }
13. }
```

What is the result?

A. Compilation fails.
B. aAaA aAa AAaa AaA
C. AAaa AaA aAa aAaA
D. AaA AAaa aAaA aAa
E. aAa AaA aAaA AAaa
F. An exception is thrown at runtime.

Answer: C

QUESTION: 237
Given:

```
5. class A {
6. void foo() throws Exception { throw new Exception(); }
7. }
8. class SubB2 extends A {
9. void foo() { System.out.println("B "); }
10. }
11. class Tester {
12. public static void main(String[] args) {
13. A a = new SubB2();
14. a.foo(); 15. }
16. }
```

What is the result?

A. B
B. B, followed by an Exception.
C. Compilation fails due to an error on line 9.
D. Compilation fails due to an error on line 14.
E. An Exception is thrown with no other output

Answer: D

QUESTION: 238
Given a method that must ensure that its parameter is not null:

```
11. public void someMethod(Object value) {
12. // check for null value
...
20. System.out.println(value.getClass());
21. }
```

What, inserted at line 12, is the appropriate way to handle a null value?

A. assert value == null;
B. assert value != null, "value is null";
C. if (value == null) { throw new AssertionException("value is null"); }
D. if (value == null) { throw new IllegalArgumentException("value is null"); }

Answer: D

QUESTION: 239
Given:

```
1. public class Mule {
2. public static void main(String[] args) {
3. boolean assert = true;
4. if(assert) {
5. System.out.println("assert is true");
6. }
7. }
8. }
```

Which command-line invocations will compile?

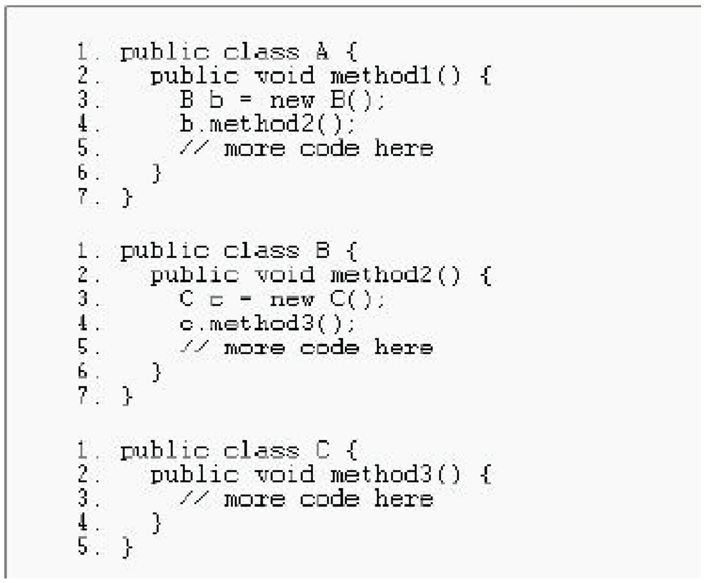
A. javac Mule.java
B. javac -source 1.3 Mule.java
C. javac -source 1.4 Mule.java
D. javac -source 1.5 Mule.java

Answer: B

QUESTION: 240
Click the Exhibit button.
Given:

```
 25. try {
26. A a = new A();
27. a.method1();
28. } catch (Exception e) {
29. System.out.print("an error occurred");
```

30. } Which two statements are true if a NullPointerException is thrown on line 3 of class C? (Choose two.)



```
1. public class A {
2.   public void method1() {
3.     B b = new B();
4.     b.method2();
5.     // more code here
6.   }
7. }

1. public class B {
2.   public void method2() {
3.     C c = new C();
4.     c.method3();
5.     // more code here
6.   }
7. }

1. public class C {
2.   public void method3() {
3.     // more code here
4.   }
5. }
```

A. The application will crash.
B. The code on line 29 will be executed.
C. The code on line 5 of class A will execute.
D. The code on line 5 of class B will execute.
E. The exception will be propagated back to line 27.

Answer: B,E

QUESTION: 241
Given:
1. public class Venus {
2. public static void main(String[] args) {
3. int [] x = {1,2,3};
4. int y[] = {4,5,6};
5. new Venus().go(x,y);
6. }
7. void go(int[]... z) {
8. for(int[] a : z)
9. System.out.print(a[0]);
10. }
11. } What is the result?

A. 1
B. 12
C. 14
D. 123
E. Compilation fails.

F. An exception is thrown at runtime.

Answer: C

QUESTION: 242

Given:

```
public class Test {
public enum Dogs {collie, harrier, shepherd};
public static void main(String [] args) {
Dogs myDog = Dogs.shepherd;
switch (myDog) {
case collie:
System.out.print("collie ");
case default:
System.out.print("retriever ");
case harrier:
System.out.print("harrier ");
}
}
}
```

What is the result?

A. harrier
B. shepherd
C. retriever
D. Compilation fails.
E. retriever harrier
F. An exception is thrown at runtime.

Answer: D

QUESTION: 243

Given:

```
static void test() {
try {
String x = null;
System.out.print(x.toString() + " ");
}
finally { System.out.print("finally "); }
}
public static void main(String[] args) {
try { test(); }
catch (Exception ex) { System.out.print("exception "); }
21. }
```
What is the result?

A. null
B. finally
C. null finally
D. Compilation fails.
E. finally exception

Answer: E

QUESTION: 244
Given:
```
1. public class Breaker2 {
2. static String o = "";
3. public static void main(String[] args) {
4. z:
5. for(int x = 2; x < 7; x++) {
6. if(x==3) continue;
7. if(x==5) break z;
8. o = o + x;
9. }
10. System.out.println(o);
11. }
12. }
```
What is the result?

A. 2
B. 24
C. 234
D. 246
E. 2346
F. Compilation fails.

Answer: B

QUESTION: 245
Given:
```
11. public static void main(String[] args) {
12. String str = "null";
13. if (str == null) {
14. System.out.println("null");
15. } else (str.length() == 0) {
16. System.out.println("zero");
17. } else {
18. System.out.println("some");
19. }
20. }
```

What is the result?

A. null
B. zero
C. some
D. Compilation fails.
E. An exception is thrown at runtime.

Answer: D

QUESTION: 246
Which can appropriately be thrown by a programmer using Java SE technology to create a desktop application?

A. ClassCastException
B. NullPointerException
C. NoClassDefFoundError
D. NumberFormatException
E. ArrayIndexOutOfBoundsException

Answer: D

QUESTION: 247
Given:

```
11. class A {
12. public void process() { System.out.print("A,"); }
13. class B extends A {
14. public void process() throws IOException {
15. super.process();
16. System.out.print("B,");
17. throw new IOException();
18. }
19. public static void main(String[] args) {
20. try { new B().process(); }
21. catch (IOException e) { System.out.println("Exception"); }
22. }
```

What is the result?

A. Exception

B. A,B,Exception
C. Compilation fails because of an error in line 20.
D. Compilation fails because of an error in line 14.
E. A NullPointerException is thrown at runtime.

Answer: D

QUESTION: 248
Given:

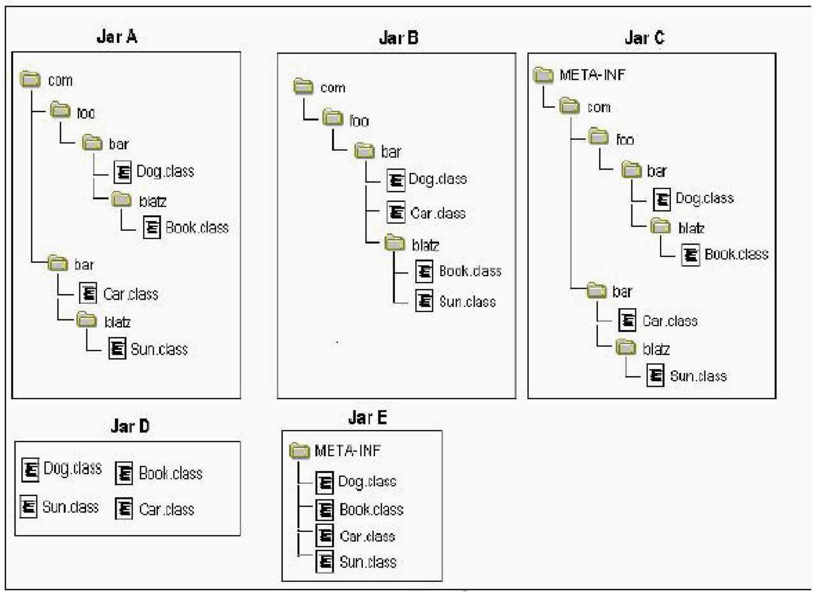
```
11. public void genNumbers() {
12. ArrayList numbers = new ArrayList();
13. for (int i=0; i<10; i++) {
14. int value = i * ((int) Math.random());
15. Integer intObj = new Integer(value);
16. numbers.add(intObj);
17. }
18. System.out.println(numbers);
19. }
```

Which line of code marks the earliest point that an object referenced by intObj becomes a candidate for garbage collection?

A. Line 16
B. Line 17
C. Line 18
D. Line 19
E. The object is NOT a candidate for garbage collection.

Answer: D

QUESTION: 249
Click the Exhibit button.
Given the fully-qualified class names:
com.foo.bar.Dog
com.foo.bar.blatz.Book
com.bar.Car
com.bar.blatz.Sun
Which graph represents the correct directory structure for a JAR file from which those classes can be used by the compiler and JVM?



A. Jar A
B. Jar B
C. Jar C
D. Jar D
E. Jar E

Answer: A

QUESTION: 250
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?

A. Line 5
B. Line 6

C. Line 7
D. Line 8
E. Line 9
F. Line 10

Answer: D

QUESTION: 251
Given:

```
15. public class Pass2 {
16. public void main(String [] args) {
17. int x = 6;
18. Pass2 p = new Pass2();
19. p.doStuff(x);
20. System.out.print(" main x = " + x);
21. }
22.
23. void doStuff(int x) {
24. System.out.print(" doStuff x = " + x++);
25. }
26. }
```

And the command-line invocations:

```
javac Pass2.
java java Pass2 5
```

What is the result?

A. Compilation fails.
B. An exception is thrown at runtime.
C. doStuff x = 6 main x = 6
D. doStuff x = 6 main x = 7
E. doStuff x = 7 main x = 6
F. doStuff x = 7 main x = 7

Answer: B

QUESTION: 252
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

QUESTION: 253
A class games.cards.Poker is correctly defined in the jar file Poker.jar. A user wants to execute the main method of Poker on a UNIX system using the command:

```
java games.cards.Poker
```

What allows the user to do this?

A. put Poker.jar in directory /stuff/java, and set the CLASSPATH to include /stuff/java
B. put Poker.jar in directory /stuff/java, and set the CLASSPATH to include /stuff/java/*.jar
C. Put Poker.jar in directory /stuff/java, and set the CLASSPATH to include /stuff/java/Poker.jar
D. put Poker.jar in directory /stuff/java/games/cards, and set the CLASSPATH to include /stuff/java E. put Poker.jar in directory /stuff/java/games/cards, and set the CLASSPATH to include /stuff/java/*.jar
F. put Poker.jar in directory /stuff/java/games/cards, and set the CLASSPATH to include /stuff/java/Poker.jar

Answer: C

QUESTION: 254
Given a correctly compiled class whose source code is:

```
1. package com.sun.sjcp;
2. public class Commander {
3. public static void main(String[] args) {
4. // more code here
5. }
6. }
```

Assume that the class file is located in /foo/com/sun/sjcp/, the current directory is /foo/, and that the classpath contains "." (current directory). Which command line correctly runs Commander?

A. java Commander
B. java com.sun.sjcp.Commander
C. java com/sun/sjcp/Commander
D. java -cp com.sun.sjcp Commander
E. java -cp com/sun/sjcp Commander

Answer: B

QUESTION: 255
Given:

```
1. interface DoStuff2 {
2. float getRange(int low, int high); }
3.
4. interface DoMore {
5. float getAvg(int a, int b, int c); }
6.
7. abstract class DoAbstract implements DoStuff2, DoMore { }
8.
9. class DoStuff implements DoStuff2 {
10. public float getRange(int x, int y) { return 3.14f; } }
11.
12. interface DoAll extends DoMore {
13. float getAvg(int a, int b, int c, int d); }
```

What is the result?

A. The file will compile without error.
B. Compilation fails. Only line 7 contains an error.
C. Compilation fails. Only line 12 contains an error.
D. Compilation fails. Only line 13 contains an error.
E. Compilation fails. Only lines 7 and 12 contain errors.
F. Compilation fails. Only lines 7 and 13 contain errors.
G. Compilation fails. Lines 7, 12, and 13 contain errors.

Answer: A

QUESTION: 256
Given:

```
3. public class Spock {
4. public static void main(String[] args) {
```

```
5. Long tail = 2000L;
6. Long distance = 1999L;
7. Long story = 1000L;
8. if((tail > distance) ^ ((story * 2) == tail))
9. System.out.print("1");
10. if((distance + 1 != tail) ^ ((story * 2) == distance))
11. System.out.print("2");
12. }
13. }
```

What is the result?

A. 1
B. 2
C. 12
D. Compilation fails.
E. No output is produced.
F. An exception is thrown at runtime.

Answer: E

QUESTION: 257
Given:

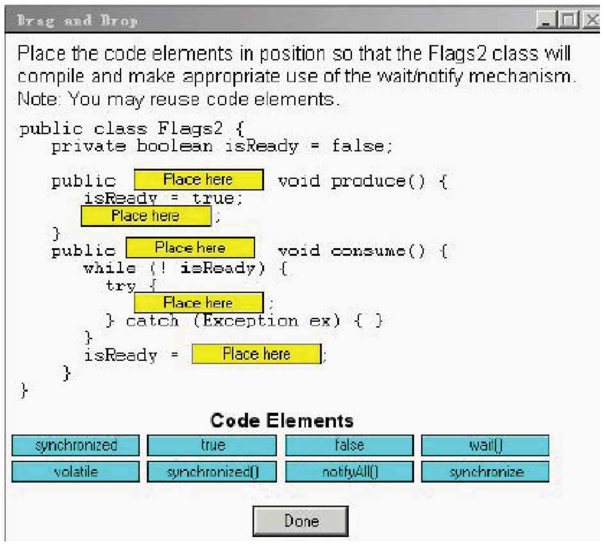
```
5.  class Payload {
6. private int weight;
7. public Payload (int w) { weight = w; }
8. public void setWeight(int w) { weight = w; }
9. public String toString() { return Integer.toString(weight); }
10. }
11. public class TestPayload {
12. static void changePayload(Payload p) { /* insert code */ }
13. public static void main(String[] args) {
14. Payload p = new Payload(200); 15. p.setWeight(1024);
16. changePayload(p);
17. System.out.println("p is " + p);
18. } }
```

Which code fragment, inserted at the end of line 12, produces the output p is 420?

A. p.setWeight(420);
B. p.changePayload(420);
C. p = new Payload(420);
D. Payload.setWeight(420);
E. p = Payload.setWeight(420);

Answer: A

QUESTION: 258
Click the Task button.



Drag and Drop

Place the code elements in position so that the Flags2 class will compile and make appropriate use of the wait/notify mechanism.
Note: You may reuse code elements.

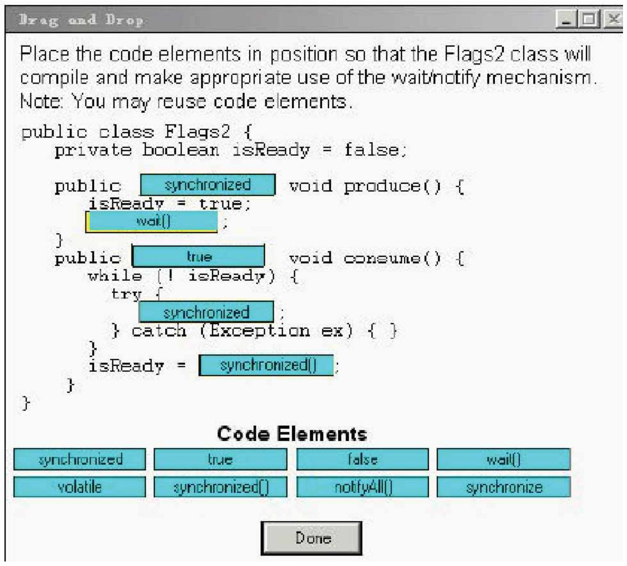
```
public class Flags2 {
    private boolean isReady = false;

    public [Place here] void produce() {
        isReady = true;
        [Place here];
    }
    public [Place here] void consume() {
        while (! isReady) {
            try {
                [Place here];
            } catch (Exception ex) { }
        }
        isReady = [Place here];
    }
}
```

Code Elements

synchronized	true	false	wait()
volatile	synchronized()	notifyAll()	synchronize

Done

Answer:



Drag and Drop

Place the code elements in position so that the Flags2 class will compile and make appropriate use of the wait/notify mechanism.
Note: You may reuse code elements.

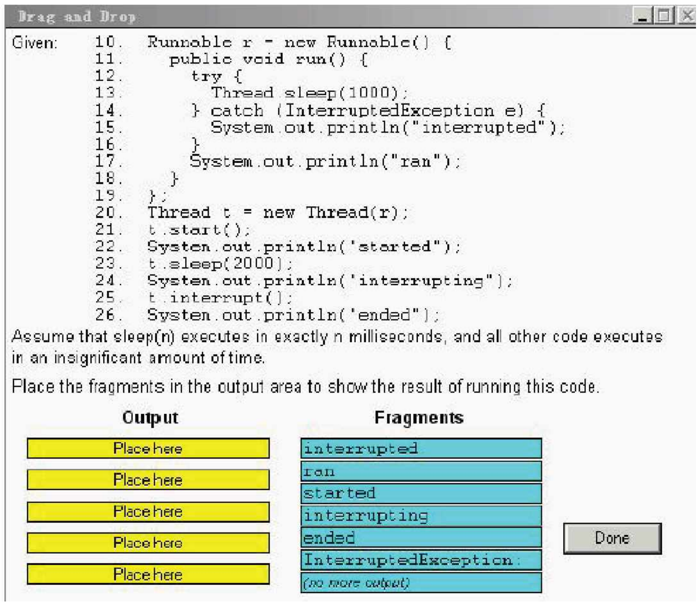
```
public class Flags2 {
    private boolean isReady = false;

    public [synchronized] void produce() {
        isReady = true;
        [wait()];
    }
    public [true] void consume() {
        while (! isReady) {
            try {
                [synchronized];
            } catch (Exception ex) { }
        }
        isReady = [synchronized()];
    }
}
```

Code Elements

synchronized	true	false	wait()
volatile	synchronized()	notifyAll()	synchronize

Done

QUESTION: 259
Click the Task button.



Given:

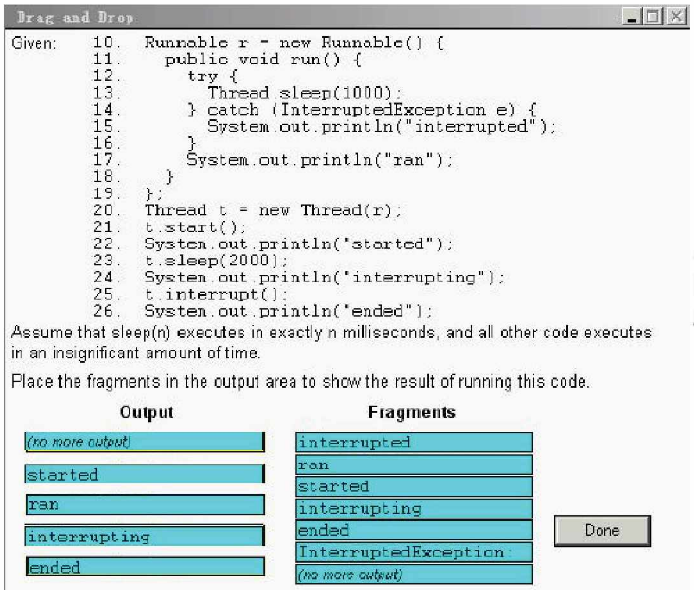
```
Runnable r = new Runnable() {
  public void run() {
    try {
      Thread.sleep(1000);
    } catch (InterruptedException e) {
      System.out.println("interrupted");
    }
    System.out.println("ran");
  }
};
Thread t = new Thread(r);
t.start();
System.out.println("started");
t.sleep(2000);
System.out.println("interrupting");
t.interrupt();
System.out.println("ended");
```

Assume that sleep(n) executes in exactly n milliseconds, and all other code executes in an insignificant amount of time.

Place the fragments in the output area to show the result of running this code.

Output	Fragments
Place here	interrupted
Place here	ran
Place here	started
Place here	interrupting
Place here	ended
	InterruptedException
	(no more output)

Answer:



Given:

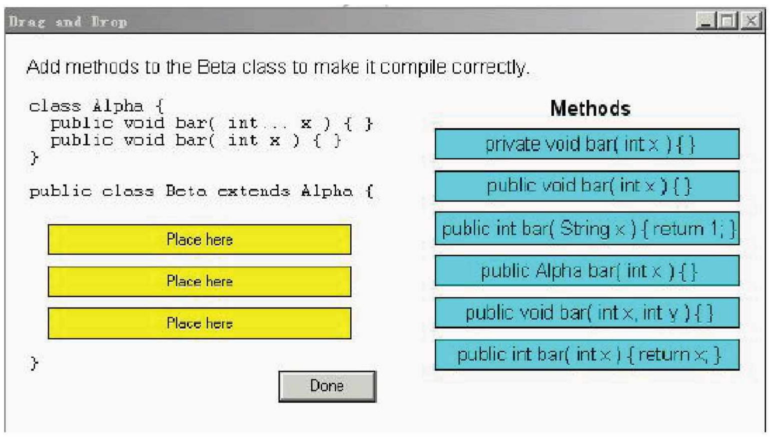
```
Runnable r = new Runnable() {
  public void run() {
    try {
      Thread.sleep(1000);
    } catch (InterruptedException e) {
      System.out.println("interrupted");
    }
    System.out.println("ran");
  }
};
Thread t = new Thread(r);
t.start();
System.out.println("started");
t.sleep(2000);
System.out.println("interrupting");
t.interrupt();
System.out.println("ended");
```

Assume that sleep(n) executes in exactly n milliseconds, and all other code executes in an insignificant amount of time.

Place the fragments in the output area to show the result of running this code.

Output	Fragments
(no more output)	interrupted
started	ran
ran	started
interrupting	interrupting
ended	ended
	InterruptedException
	(no more output)

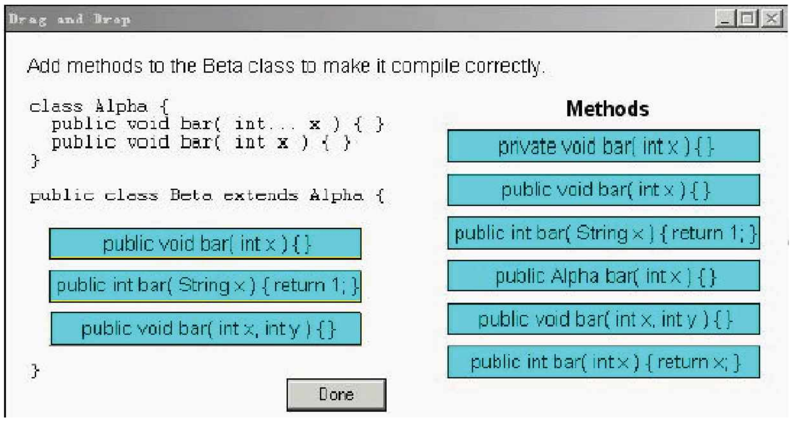
QUESTION: 260
Click the Task button.


Add methods to the Beta class to make it compile correctly.

```
class Alpha {
   public void bar( int ... x ) { }
   public void bar( int x ) { }
}

public class Beta extends Alpha {

   Place here
   Place here
   Place here

}
```

Drag and Drop
Methods
private void bar(int x) { }
public void bar(int x) { }
public int bar(String x) { return 1; }
public Alpha bar(int x) { }
public void bar(int x, int y) { }
public int bar(int x) { return x; }
Done

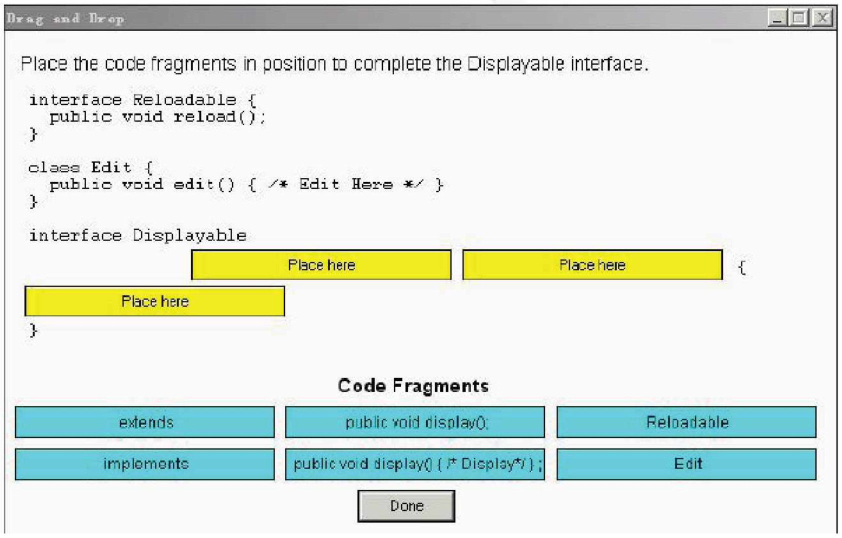
Answer:


Add methods to the Beta class to make it compile correctly.

```
class Alpha {
   public void bar( int ... x ) { }
   public void bar( int x ) { }
}

public class Beta extends Alpha {

   public void bar( int x ) { }
   public int bar( String x ) { return 1; }
   public void bar( int x, int y ) { }

}
```

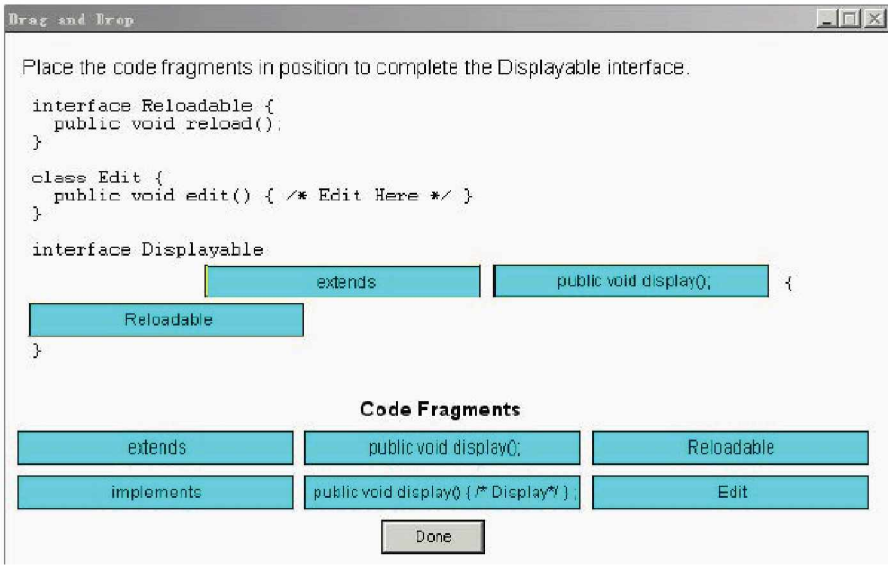
Drag and Drop
Methods
private void bar(int x) { }
public void bar(int x) { }
public int bar(String x) { return 1; }
public Alpha bar(int x) { }
public void bar(int x, int y) { }
public int bar(int x) { return x; }
Done

QUESTION: 261
Click the Task button.


Place the code fragments in position to complete the Displayable interface.

```
interface Reloadable {
   public void reload();
}

class Edit {
   public void edit() { /* Edit Here */ }
}

interface Displayable
                    Place here      Place here      {
Place here
}
```

Answer:



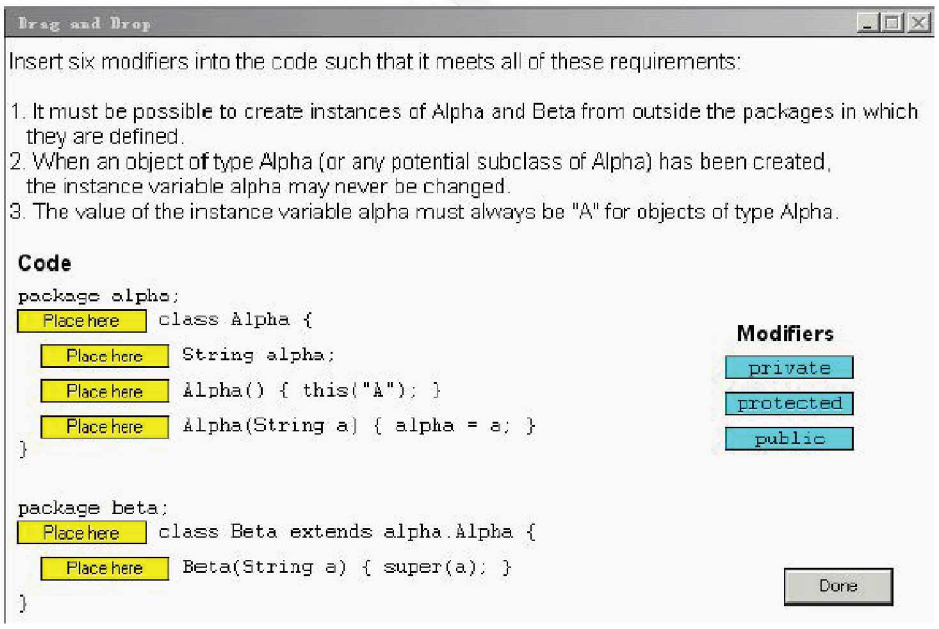
Place the code fragments in position to complete the Displayable interface.

```
interface Reloadable {
  public void reload();
}

class Edit {
  public void edit() { /* Edit Here */ }
}

interface Displayable
                extends          public void display();   {
Reloadable
}
```

Code Fragments

extends	public void display();	Reloadable
implements	public void display() { /* Display*/ }	Edit

QUESTION: 262
Click the Task button.



Insert six modifiers into the code such that it meets all of these requirements:

1. It must be possible to create instances of Alpha and Beta from outside the packages in which they are defined.
2. When an object of type Alpha (or any potential subclass of Alpha) has been created, the instance variable alpha may never be changed.
3. The value of the instance variable alpha must always be "A" for objects of type Alpha.

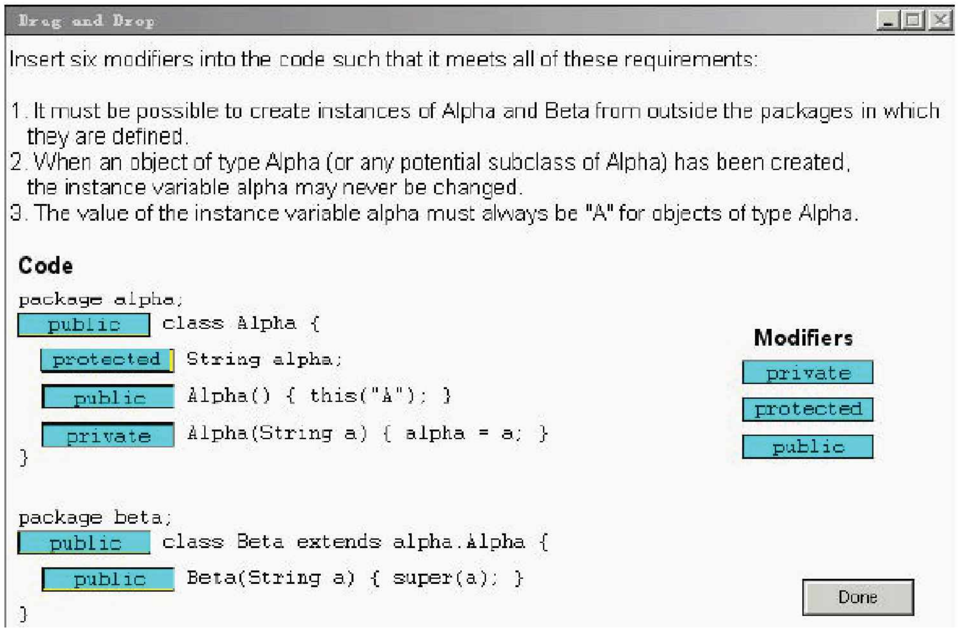
Code

```
package alpha;
Place here class Alpha {
  Place here String alpha;
  Place here Alpha() { this("A"); }
  Place here Alpha(String a) { alpha = a; }
}

package beta;
Place here class Beta extends alpha.Alpha {
  Place here Beta(String a) { super(a); }
}
```

Modifiers

private
protected
public

Answer:



Drag and Drop

Insert six modifiers into the code such that it meets all of these requirements:

1. It must be possible to create instances of Alpha and Beta from outside the packages in which they are defined.
2. When an object of type Alpha (or any potential subclass of Alpha) has been created, the instance variable alpha may never be changed.
3. The value of the instance variable alpha must always be "A" for objects of type Alpha.

Code

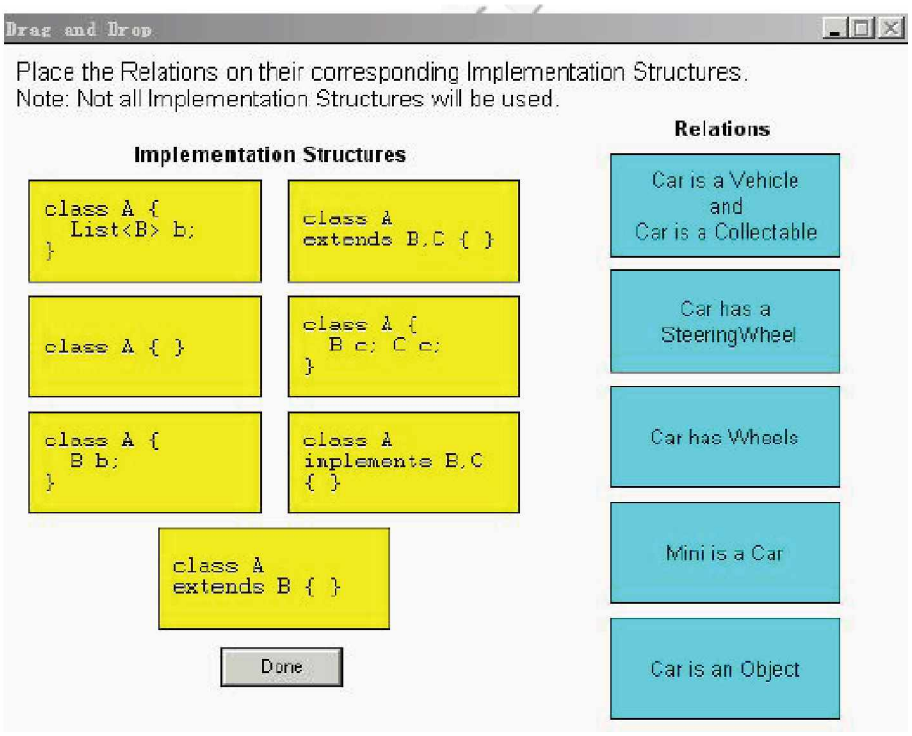
```
package alpha;
public class Alpha {
  protected String alpha;
  public Alpha() { this("A"); }
  private Alpha(String a) { alpha = a; }
}

package beta;
public class Beta extends alpha.Alpha {
  public Beta(String a) { super(a); }
}
```

Modifiers

private
protected
public

Done

QUESTION: 263
Click the Task button.



Drag and Drop

Place the Relations on their corresponding Implementation Structures.
Note: Not all Implementation Structures will be used.

Implementation Structures

```
class A {
  List<B> b;
}
```

```
class A
extends B,C { }
```

```
class A { }
```

```
class A {
  B c; C c;
}
```

```
class A {
  B b;
}
```

```
class A
implements B,C
{ }
```

```
class A
extends B { }
```

Done

Relations

Car is a Vehicle and Car is a Collectable

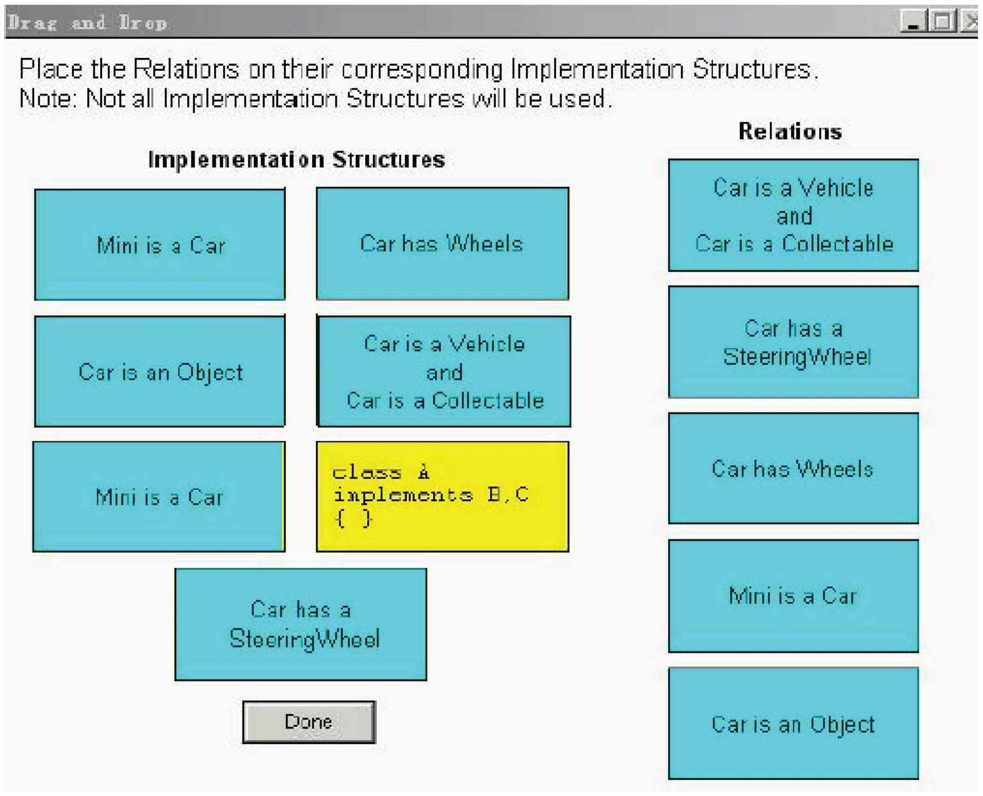
Car has a SteeringWheel

Car has Wheels

Mini is a Car

Car is an Object

Answer:


Drag and Drop

Place the Relations on their corresponding Implementation Structures.
Note: Not all Implementation Structures will be used.

Implementation Structures

Mini is a Car

Car has Wheels

Car is an Object

Car is a Vehicle and Car is a Collectable

Mini is a Car

```
class A
implements B,C
{ }
```

Car has a SteeringWheel

Done

Relations

Car is a Vehicle and Car is a Collectable

Car has a SteeringWheel

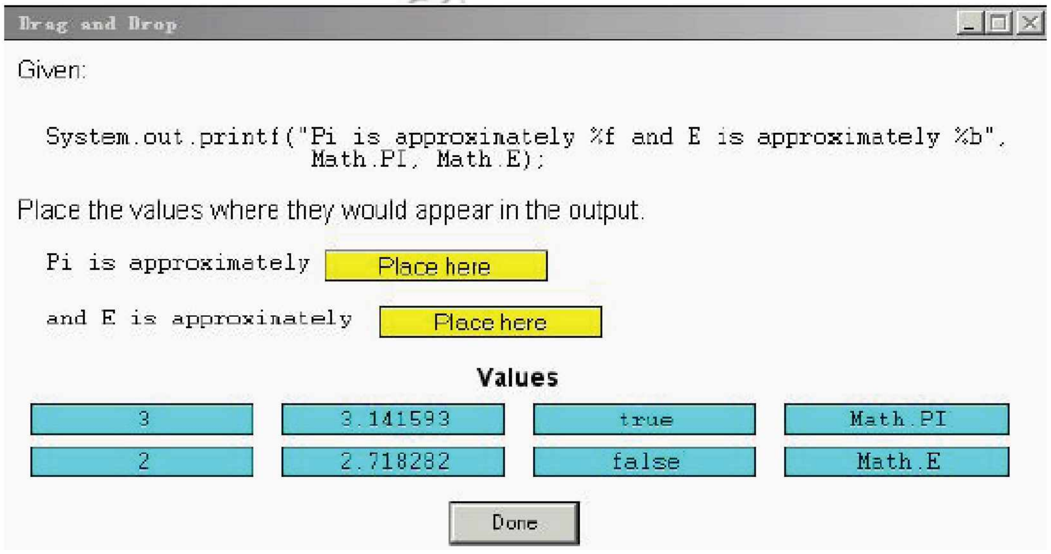
Car has Wheels

Mini is a Car

Car is an Object

QUESTION: 264

Click the Task button.


Drag and Drop

Given:

```
System.out.printf("Pi is approximately %f and E is approximately %b",
                   Math.PI, Math.E);
```

Place the values where they would appear in the output.

Pi is approximately Place here

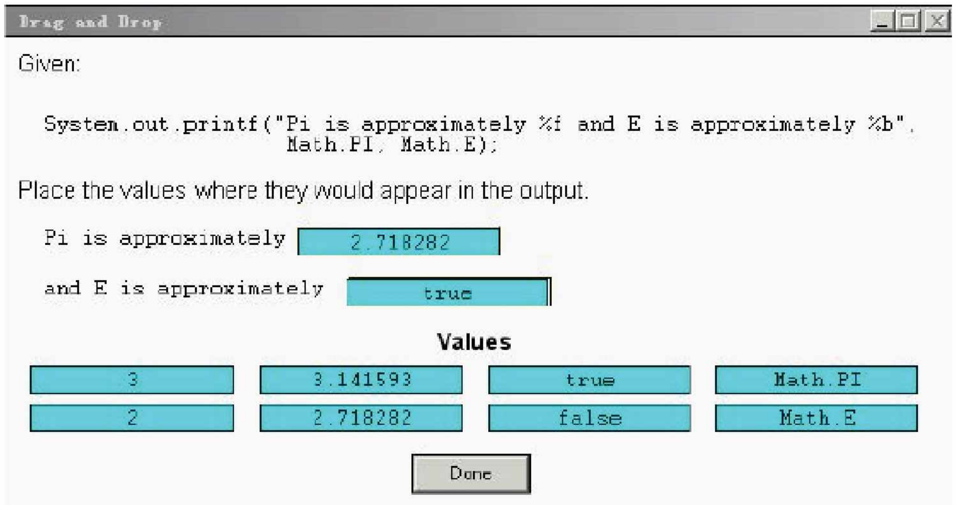
and E is approximately Place here

Values

3	3.141593	true	Math.PI
2	2.718282	false	Math.E

Done

Answer:



Given:

```
System.out.printf("Pi is approximately %f and E is approximately %b",
                  Math.PI, Math.E);
```

Place the values where they would appear in the output.

Pi is approximately 2.718282

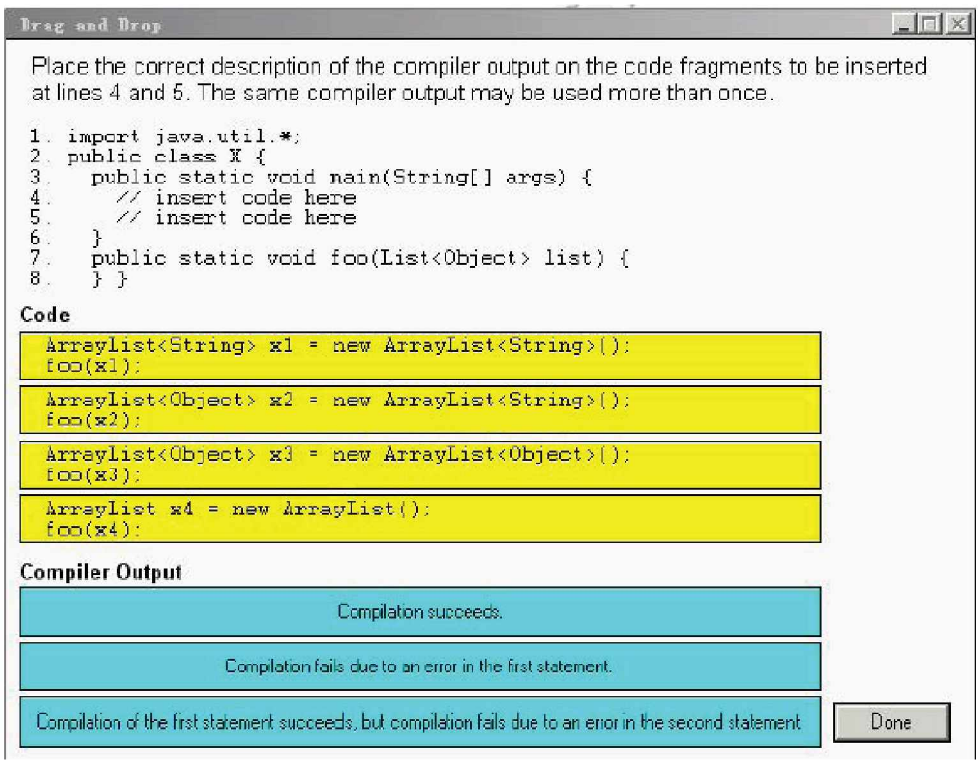
and E is approximately true

Values

3	3.141593	true	Math.PI
2	2.718282	false	Math.E

QUESTION: 265

Click the Task button.



Place the correct description of the compiler output on the code fragments to be inserted at lines 4 and 5. The same compiler output may be used more than once.

```
1. import java.util.*;
2. public class X {
3.   public static void main(String[] args) {
4.     // insert code here
5.     // insert code here
6.   }
7.   public static void foo(List<Object> list) {
8.   } }
```

Code

```
Arraylist<String> x1 = new ArrayList<String>();
foo(x1);
```

```
Arraylist<Object> x2 = new ArrayList<String>();
foo(x2);
```

```
Arraylist<Object> x3 = new ArrayList<Object>();
foo(x3);
```

```
Arraylist x4 = new ArrayList();
foo(x4);
```

Compiler Output

Compilation succeeds.

Compilation fails due to an error in the first statement.

Compilation of the first statement succeeds, but compilation fails due to an error in the second statement

Answer: