

Q : 1 Given:

```
11. public class Person {
12.     private String name, comment;
13.     private int age;
14.     public Person(String n, int a, String c) {
15.         name = n; age = a; comment = c;
16.     }
17.     public boolean equals(Object o) {
18.         if (! (o instanceof Person)) return false;
19.         Person p = (Person)o;
20.         return age == p.age && name.equals(p.name);
21.     }
22. }
```

What is the appropriate definition of the hashCode method in class Person?

- A. return super.hashCode();
- B. return name.hashCode() + age * 7;
- C. return name.hashCode() + comment.hashCode() / 2;
- D. return name.hashCode() + comment.hashCode() / 2 - age * 3;

Answer: B

Q : 2 34. HashMap props = new HashMap();
35. props.put("key45", "some value");
36. props.put("key12", "some other value");
37. props.put("key39", "yet another value");
38. Set s = props.keySet();
39. // insert code here

What, inserted at line 39, will sort the keys in the props HashMap?

- A. Arrays.sort(s);
- B. s = new TreeSet(s);
- C. Collections.sort(s);
- D. s = new SortedSet(s);

Answer: B

Q : 3 Which statement is true about the set variable on line 12?

```
1. import java.util.*;
2. public class TestSet {
3.     enum Example { ONE, TWO, THREE }
4.     public static void main(String[] args)
5.     {
6.         Collection coll = new ArrayList();
7.         coll.add(Example.THREE);
8.         coll.add(Example.THREE);
9.         coll.add(Example.THREE);
10.        coll.add(Example.TWO);
11.        coll.add(Example.TWO);
12.        Set set = new HashSet(coll);
13.    }
14. }
```

- A. The set variable contains all six elements from the coll collection, and the order is guaranteed to be preserved.
- B. The set variable contains only three elements from the coll collection, and the order is guaranteed to be preserved.

- C. The set variable contains all six elements from the coll collection, but the order is NOT guaranteed to be preserved.
- D. The set variable contains only **three elements** from the coll collection, but the order is **NOT** guaranteed to be preserved.

Answer: D

Q : 4 23. Object [] myObjects = {
24. new Integer(12),
25. new String("foo"),
26. new Integer(5),
27. new Boolean(true)
28. };
29. Arrays.sort(myObjects);
30. for(int i=0; i<myObjects.length; i++) {
31. System.out.print(myObjects[i].toString());
32. System.out.print(" ");
33. }

What is the result?

- A. Compilation fails due to an error in line 23.
- B. Compilation fails due to an error in line 29.
- C. **A ClassCastException occurs in line 29.**
- D. A ClassCastException occurs in line 31.
- E. The value of all four objects prints in natural order.

Answer: C

Q: 5 Click the Task button

Place code into the class so that it compiles and generates the output answer=42. Note: Code options may be used more than once.

Class

```
public class Gen<T> {  
    private T object;  
    public Gen (T object) {  
        this.object = object;  
    }  
    public T getObject() {  
        return object;  
    }  
  
    public static void main(String[] args) {  
        Gen<String> str = new Gen<String>("answer");  
        Gen<Integer> intg = new Gen<Integer>(42);  
        System.out.println(str.getObject() + "=" +  
            intg.getObject());  
    }  
}
```

Code Options

Gen<T>

Gen<?>

Gen

?

T

Done

Q: 6 Click the Task button.

Given:

```
public void takeList(List<? extends String> list) {  
    // insert code here  
}
```

Place the Compilation Results on each code statement to indicate whether or not that code will compile if inserted into the takeList() method.

Code Statements

list.add("Foo");

fails

list = new ArrayList<String>();

succeeds

list = new ArrayList<Object>();

fails

String s = list.get(0);

succeeds

Object o = list;

succeeds

Compilation Results

Compilation succeeds

Compilation fails

Done

Q: 7 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);

(iii)

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

(ii)

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

(i)

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

(i)

Compiler Output

(i). Compilation succeeds.

(ii). Compilation fails due to an error in the first statement.

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

Done

Q: 8 Given:

```
1. public class Person{
2. private String name;
3. public Person(String name) { this.name = name; }
4. public boolean equals(Person p) {
5. return p.name.equals(this.name);
6. }
7. }
```

Which statement is true?

- A. The equals method does NOT properly override the Object.equals method.
- B. Compilation fails because the private attribute p.name cannot be accessed in line 5.
- C. To work correctly with hash-based data structures, this class must also implement the hashCode method.
- D. When adding Person objects to a java.util.Set collection, the equals method in line 4 will prevent duplicates.

Answer: A

Q: 9 Given:

```
1. import java.util.*;
2. public class Old{
```

```

3.public static Object get0(List list){
4.return list.get(0);
5.}
6.}

```

Which three will compile successfully? (Choose three.)

- A. Object o= Old.get0(new LinkedList());
- B. Object o = Old.get0(new LinkedList<?>());
- C. String s = Old.get0(new LinkedList<String>());
- D. Object o = Old.get0(new LinkedList<Object>());
- E. String s = (String)Old.get0(new LinkedList<String>());

Answer: A, D, E

Q: 10 Given:

```

1. import java.util.*;
2. public class Example {
3.public static void main(String[] args){
4.// insert code here
5.set.add(new Integer(2));
6.set.add(new Integer(1));
7.System.out.println(set);
8.}
9.}

```

Which code, inserted at line 4, guarantees that this program will output [1, 2]?

- A. Set set = new TreeSet();
- B. Set set = new HashSet();
- C. Set set = new SortedSet();
- D. List set = new SortedList();
- E. Set set = new LinkedHashSet();

Answer: A

Q: 11 Given:

```

10. class Foo {
11.static void alpha(){/* more code here */ }
12.void beta(){/* more code here */ }
13.}

```

Which two statements are true? (Choose two.)

- A. Foo.beta() is a valid invocation of beta().
- B. Foo.alpha() is a valid invocation of alpha().
- C. Method beta() can directly call method alpha().
- D. Method alpha() can directly call method beta().

Answer: B, C

Q: 12 Click the Task button.

Place the Output Options in the Actual Output Sequence to indicate the output from this code:

```
class Alpha {
    public void foo( String... args )
    { System.out.print("Alpha:foo "); }
    public void bar( String a )
    { System.out.print("Alpha:bar "); }
}

public class Beta extends Alpha {
    public void foo( String a )
    { System.out.print("Beta:foo "); }
    public void bar( String a )
    { System.out.print("Beta:bar "); }
    public static void main( String[] argv ) {
        Alpha a = new Beta();
        Beta b = (Beta)a;
        a.foo( "test" ); b.foo( "test" );
        a.bar( "test" ); b.bar( "test" );
    }
}
```

Actual Output Sequence

Alpha:foo Beta:foo Beta:bar Beta:bar

Output Options

Alpha:foo Alpha:bar Beta:foo Beta:bar

Done

Q: 13 Given:

```
11. public static void parse(String str){
12. try{
13. float f= Float.parseFloat(str);
14. } catch(NumberFormatException nfe) {
15. f= 0;
16. } finally {
17. System.out.println(f);
18. }
19. }
20. public static void main(String[] args){
21. parse("invalid");
22. }
```

What is the result?

- A. 0.0
- B. Compilation fails.
- C. A ParseException is thrown by the parse method at runtime.
- D. A NumberFormatException is thrown by the parse method at runtime.

Answer: B

Q: 14 Given:

```
10. class Line {
11. public static class Point{}
```



```

12.}
13.
14. class Triangle {
15.// insert code here
16.}

```

Which code, inserted at line 15, creates an instance of the Point class defined in Line?

- A. Point p = new Point();
- B. Line.Point p = new Line.Point();
- C. The Point class cannot be instantiated at line 15.
- D. Line l = new Line() ; l.Point p = new l.Point();

Answer: B

Q: 15 Click the Task button.

Place the lines in the correct order to complete the enum.

```
enum Element {
```

- | | |
|-----|-------|
| 1st | (iv) |
| 2nd | (iii) |
| 3rd | (ii) |
| 4th | (i) |
| 5th | (v) |

Lines

- (i). `public String info() { return "element"; }`
- (ii). `};`
- (iii). `FIRE { public String info() { return "Hot"; }`
- (iv). `EARTH, WIND,`
- (v). `}`

Q: 16 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: 17 Given:

```

10. public class Bar {
11. static void foo( int... x){
12. // insert code here
13. }
14. }

```

Which two code fragments, inserted independently at line 12, will allow the class to compile? (Choose two.)

- A. foreach(x) System.out.println(z);
- B. for(int z : x) System.out.println(z);
- C. while(x.hasNext()) System.out.println(x.next());
- D. for(int i=0; i< x.length; i++) System.out.println(x[i]);

Answer: B, D

Q: 18 Given

```

1. public interface A {
2.     public void doSomething(String thing);
3. }

1. public class AImpl implements A {
2.     public void doSomething(String msg) { }
3. }

1. public class B {
2.     public A doit() {
3.         // more code here
4.     }
5.
6.     public String execute() {
7.         // more code here
8.     }
9. }

```



```

1. public class C extends B {
2.     public AImpl doit() {
3.         // more code here
4.     }
5.
6.     public Object execute() {
7.         // more code here
8.     }
9. }

```

Which statement is true about the classes and interfaces in the exhibit?

- A. Compilation will succeed for all classes and interfaces.
- B. Compilation of class C will fail because of an error in line 2.
- C. Compilation of class C will fail because of an error in line 6.
- D. Compilation of class AImpl will fail because of an error in line 2.

Answer: C

Q: 19 Given:

```

1. public class Plant {
2. private String name;
3. public Plant(String name) { this.name = name; }
4. public String getName() { return name; }
5. }
1. public class Tree extends Plant{
2. public void growFruit() { }
3. public void dropLeaves() { }
4. }

```

Which statement is true?

- A. The code will compile without changes.
- B. The code will compile if public Tree() { Plant(); } is added to the Tree class.
- C. The code will compile if public Plant() { Tree(); } is added to the Plant class.
- D. The code will compile if public Plant() { this("fern"); } is added to the Plant class.
- E. The code will compile if public Plant() { Plant("fern"); } is added to the Plant class.

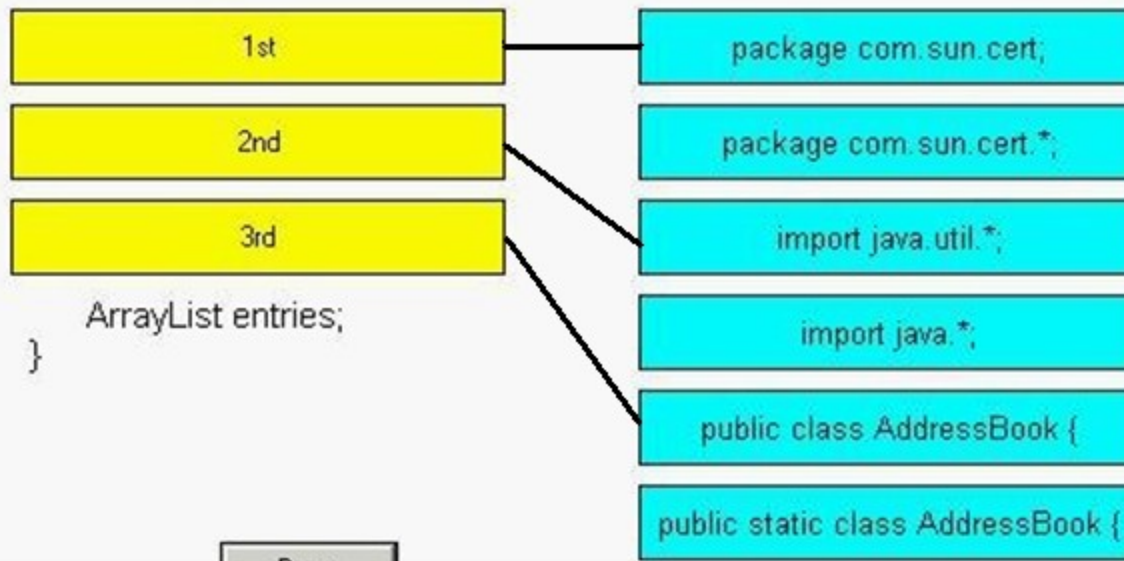
Answer: D

Q: 20 Click the Task button.

Place the code elements in order so that the resulting Java source file will compile correctly, resulting in a class called com.sun.cert.AddressBook.

Source File

Code Element



Q: 21 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
```

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

Code Fragments

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

Q: 22 Which two classes correctly implement both the java.lang.Runnable and the java.lang.Cloneable interfaces? (Choose two.)

- A. public class Session
implements Runnable, Cloneable {
public void run();
public Object clone();
}
- B. public class Session
extends Runnable, Cloneable {

- ```
public void run() { /* do something */ }

public Object clone() { /* make a copy */ }
```
- C. public class Session  
implements Runnable, Cloneable {  
public void run() { /\* do something \*/ }  
public Object clone() { /\* make a copy \*/ }
- D. public abstract class Session  
implements Runnable, Cloneable {  
public void run() { /\* do something \*/ }  
public Object clone() { /\* make a copy \*/ }
- E. public class Session  
implements Runnable, implements Cloneable {  
public void run() { /\* do something \*/ }  
public Object clone() { /\* make a copy \*/ }

Answer: C, D

Q: 23 Click the Exhibit button.

```
1. class Computation extends Thread {
2.
3. private int num;
4. private boolean isComplete;
5. private int result;
6.
7. public Computation(int num) { this.num
= num; }
8.
9. public synchronized void run() {
10. result = num * 2;
11. isComplete = true;
12. notify();
13. }
14.
15. public synchronized int getResult() {
16. while (!isComplete) {
17. try {
18. wait();
19. } catch (InterruptedException e)
{}
20. }
21. return result;
22. }
23.
```

```

24. public static void main(String[] args)
 {
25. Computation[] computations = new
Computation[4];
26. for (int i = 0; i <
computations.length; i++) {
27. computations[i] = new
Computation(i);
28. computations[i].start();
29. }
30. for (Computation c : computations)
31. System.out.print(c.getResult() + "
");
32. }
33. }

```

What is the result?

- A. The code will deadlock.
- B. The code may run with no output.
- C. An exception is thrown at runtime.
- D. The code may run with output "0 6".
- E. The code may run with output "2 0 6 4".
- F. The code may run with output "0 2 4 6".

Answer: F

Q: 24 Given:

```

1. public class Threads2 implements Runnable {
2.
3. public void run() {
4. System.out.println("run.");
5. throw new RuntimeException("Problem");
6. }
7. public static void main(String[] args){
8. Thread t = new Thread(new Threads2());
9. t.start();
10. System.out.println("End of method.");
11. }
12. }

```

Which two can be results? (Choose two.)

- A. java.lang.RuntimeException: Problem
- B. run.java.lang.RuntimeException: Problem
- C. End of method.java.lang.RuntimeException: Problem
- D. End of method.run.java.lang.RuntimeException: Problem
- E. run.java.lang.RuntimeException: ProblemEnd of method.

Answer: D, E

Q: 25 Given:

```

1. public class TestSeven extends Thread {
2. private static int x;
3. public synchronized void doThings() {
4. int current= x;
5. current++;
6. x = current;
7. }

8. public void run() {
9. doThings();
10. }
11. }

```

Which statement is true?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. Synchronizing the run() method would make the class thread-safe.
- D. The data in variable "x" are protected from concurrent access problems.
- E. Declaring the doThings() method as static would make the class thread-safe.
- F. Wrapping the statements within doThings() in a synchronized(new Object()){ } block would make the class thread-safe.

Answer: E

**Q: 26 Given:**

```

1. public class Threads3 implements Runnable {
2. public void run() {
3. System.out.print("running");
4. }
5. public static void main(String[] args) {
6. Thread t = new Thread(new Threads3());
7. t.run();
8. t.run();
9. t.start();
10. }
11. }

```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes and prints "running".
- D. The code executes and prints "runningrunning".
- E. The code executes and prints "runningrunningrunning".

Answer: E

**Q: 27 Given:**

```

public class NamedCounter {
private final String name;
private int count;

```

```

public NamedCounter(String name)
{ this.name = name; }
public String getName() { return name; }
public void increment() { count++; }
public int getCount(){ return count; }
public void reset(){ count = 0; }
}

```

Which three changes should be made to adapt this class to be used safely by multiple threads? (Choose three.)

- A. declare reset() using the synchronized keyword
- B. declare getName() using the synchronized keyword
- C. declare getCount() using the synchronized keyword
- D. declare the constructor using the synchronized keyword
- E. declare increment() using the synchronized keyword

Answer: A, C, E

Q: 28 Given:

```

7.void waitForSignal() {
8.Object obj = new Object();
9.synchronized (Thread.currentThread()) {
10.obj.wait();
11.obj.notify();
12.}
13.}

```

Which statement is true?

- A. This code may throw an InterruptedException.
- B. This code may throw an IllegalStateException.
- C. This code may throw a TimeoutException after ten minutes.
- D. This code will not compile unless "obj.wait()" is replaced with "((Thread) obj).wait()".
- E. Reversing the order of obj.wait() and obj.notify() may cause this method to complete normally.
- F. A call to notify() or notifyAll() from another thread may cause this method to complete normally.

Answer: B

Q: 29 Which two code fragments will execute the method doStuff() in a separate thread? (Choose two.)

- A. new Thread() {  
    public void run(){ doStuff(); }  
};
- B. new Thread() {  
    public void start() { doStuff(); }  
};
- C. new Thread() {  
    public void start(){ doStuff(); }  
};



```

 }.run();
D. new Thread() {
 public void run(){ doStuff(); }
 }.start();
E. new Thread(new Runnable() {
 public void run(){ doStuff(); }
 }).run();
F. new Thread(new Runnable() {
 public void run(){ doStuff(); }
 }).start();

```

Answer: D, F

Q: 30 Given:

```

1. public class TestOne implements Runnable {
2. public static void main (String[] args) throws Exception {
3. Thread t = new Thread(new TestOne());
4. t.start();
5. System.out.print("Started");
6. t.join();
7. System.out.print("Complete");
8. }
9. public void run() {
10. for (int i = 0; i < 4; i++) {
11. System.out.print(i);
12. }
13. }
14. }

```

What can be a result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes and prints "StartedComplete".
- D. The code executes and prints "StartedComplete0123".
- E. The code executes and prints "Started0123Complete".

Answer: E

**Q: 31 Given:**

```

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

```

```
19.System.out.print("retriever ");
20.case harrier:
21.System.out.print("harrier ");
22.}
23.}
24.}
```

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

Q: 32 Given:

```
8. public class test{
9.public static void main(String [] a){
10.assert a.length== 1;
11.}
12.}
```

Which two will produce an AssertionError? (Choose two.)

- A. java test
- B. java -ea test
- C. java test file1
- D. java -ea test file1
- E. java -ea test file1 file2
- F. java -ea:test test file1

Answer: B, E

Q: 33 Given:

```
10. interface Foo {}
11. class Alpha implements Foo {}
12. class Beta extends Alpha {}
13. class Delta extends Beta {
14.public static void main(String[] args){
15.Beta x = new Beta();
16.// insert code here
17.}
18.}
```

Which code, inserted at line 16, will cause a java.lang.ClassCastException?

- A. Alpha a = x;
- B. Foo f = (Delta)x;

- C. Foo f = (Alpha)x;
- D. Beta b = (Beta)(Alpha)x;

Answer: B

Q: 34 Given:

```
11. public static Collection get(){
12. Collection sorted= new LinkedList();
13. sorted.add("B"); sorted.add("C"); sorted.add("A");
14. return sorted;
15.}
16. public static void main(String[] args){
17. for (Object obj: get()){
18. System.out.print(obj+ ", ");
19.}
20.}
```

What is the result?

- A. A, B, C,
- B. B, C, A,
- C. Compilation fails.
- D. The code runs with no output.
- E. An exception is thrown at runtime.

Answer: B

Q: 35 Given:

```
84. try {
85. ResourceConnection con = resourceFactory.getConnection();
86. Results r = con.query("GET INFO FROM CUSTOMER");
87. info= r.getData();
88. con.close();
89.} catch (ResourceException re) {
90. errorLog.write(re.getMessage());
91.}
92. return info;
```

Which statement is true if a ResourceException is thrown on line 86?

- A. Line 92 will not execute.
- B. The connection will not be retrieved in line 85.
- C. The resource connection will not be closed on line 88.
- D. The enclosing method will throw an exception to its caller.

Answer: C

Q: 36 Given:

```
31.// some code here
32. try {
```

```

33.// some code here
34.} catch (SomeException se) {
35.// some code here
36.} finally{
37.// some code here
38.}

```

Under which three circumstances will the code on line 37 be executed? (Choose three.)

- A. The instance gets garbage collected.
- B. The code on line 33 throws an exception.
- C. The code on line 35 throws an exception.
- D. The code on line 31 throws an exception.
- E. The code on line 33 executes successfully.

Answer: B, C, E

Q: 37 Click the Task button.

Place the correct Code in the Code Sample to achieve the expected results.

### Expected Results

Output: 1 2 4 8 16 32

### Code Sample

```

int [] y = { 1, 2, 4, 8, 16, 32 };
System.out.print("Output: ");
for(int x : y) {
 System.out.print(x);
 System.out.print(" ");
}

```

### Code

|                            |                       |
|----------------------------|-----------------------|
| for(int x : y) {           | for(int x = y[]) {    |
| foreach (y as x) {         | foreach (int x : y) { |
| for(int x=1; x=y[]; x++) { |                       |

Q: 38 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"); }}
```

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

**Q: 39** 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 AssertionError("value is null"); }
- D. if (value == null) {throw new IllegalArgumentException("value is null"); }

Answer: D

**Q: 40** Given:

```
11. static void test() throws Error {
12.if(true) throw new AssertionError();
13.System.out.print("test ");
14.}
15. public static void main(String[] args){
16.try{ test();}
17.catch (Exception ex) { System.out.print("exception "); }
18.System.out.print("end ");
19.}
```

What is the result?

- A. end
- B. Compilation fails.
- C. exception end
- D. exception test end
- E. A Throwable is thrown by main.
- F. An Exception is thrown by main.

Answer: E

**Q: 41** Given:

```
11. Float pi= new Float(3.14f);
```

```

12. if(pi > 3) {
13. System.out.print("pi is bigger than 3. ");
14. }
15. else{
16. System.out.print("pi is not bigger than 3. ");
17. }
18. finally{
19. System.out.println("Have a nice day.");
20. }

```

What is the result?

- A. Compilation fails.
- B. pi is bigger than 3.
- C. An exception occurs at runtime.
- D. pi is bigger than 3. Have a nice day.
- E. pi is not bigger than 3. Have a nice day.

Answer: A

Q: 42 Click the Task button.

Chain these constructors to create objects to read from a file named "in" and to write to a file named "out."

```

reader = new BufferedReader(new FileReader("in"));
writer = new PrintWriter(new BufferedWriter(new FileWriter("out")));

```

#### Constructors

|                     |                  |                     |
|---------------------|------------------|---------------------|
| new FileReader(     | new PrintReader( | new BufferedReader( |
| new BufferedWriter( | new FileWriter(  | new PrintWriter(    |

Q: 43 Given:

```

11. String test = "This is a test";
12. String[] tokens = test.split("\s");
13. System.out.println(tokens.length);

```

What is the result?

- A. 0
- B. 1
- C. 4
- D. Compilation fails.
- E. An exception is thrown at runtime.

Answer: D

Q: 44 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?

- A. int Long
- B. Short Long
- C. Compilation fails.
- D. An exception is thrown at runtime.

Answer: A

Q: 45 Given:

```
12. System.out.format("Pi is approximately %d.", Math.PI);
```

What is the result?

- A. Compilation fails.
- B. Pi is approximately 3.
- C. Pi is approximately 3.141593.
- D. An exception is thrown at runtime.

Answer: D

Q: 46 Given:

```

33. Date d = new Date(0);
34. String ds = "December 15, 2004";
35.// insert code here
36. try{
37.d = df.parse(ds);
38.}
39. catch(ParseException e) {
40.System.out.println("Unable to parse " + ds);
41.}
42.// insert code here too

```

What creates the appropriate DateFormat object and adds a day to the Date object?

- A.35. DateFormat df = DateFormat.getDateFormat();  
42. d.setTime((60 \* 60 \* 24) + d.getTime());
- B.35. DateFormat df = DateFormat.getDateInstance();  
42. d.setTime((1000 \* 60 \* 60 \* 24) + d.getTime());
- C. 35. DateFormat df = DateFormat.getDateFormat();

```
42. d.setLocalTime((1000*60*60*24) + d.getLocalTime());
D. 35. DateFormat df = DateFormat.getDateInstance();
42. d.setLocalTime((60 * 60 * 24) + d.getLocalTime());
```

Answer: B

Q: 47 Given:

```
12. NumberFormat nf = NumberFormat.getInstance();
13. nf.setMaximumFractionDigits(4);
14. nf.setMinimumFractionDigits(2);
15. String a = nf.format(3.1415926);
16. String b = nf.format(2);
```

Which two statements are true about the result if the default locale is Locale.US? (Choose two.)

- A. The value of b is 2.
- B. The value of a is 3.14.
- C. The value of b is 2.00.
- D. The value of a is 3.141.
- E. The value of a is 3.1415.
- F. The value of a is 3.1416.
- G. The value of b is 2.0000.

Answer: C, F

Q: 48 Given:

```
12. import java.io.*;
13. public class Forest implements Serializable {
14. private Tree tree = new Tree();
15. public static void main(String [] args){
16. Forest f = new Forest();
17. try{
18. FileOutputStream fs = new FileOutputStream("Forest.ser");
19. ObjectOutputStream os = new ObjectOutputStream(fs);
20. os.writeObject(f); os.close();
21. } catch(Exception ex) { ex.printStackTrace();}
22. }
23.
24. class Tree {}
```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. An instance of Forest is serialized.
- D. An instance of Forest and an instance of Tree are both serialized.

Answer: B

Q: 49 Click the Task button.

Place the code fragments into position to use a BufferedReader to read in an entire text file.

```
class PrintFile {
 public static void main(String[] args){
 BufferedReader buffReader = null;
 //more code here to initialize buffReader
 try {
 String temp;

 while((temp = buffReader.readLine()) != null) {
 System.out.println(temp);
 }
 } catch (IOException e) {
 e.printStackTrace();
 }
 }
}
```

### Code Fragments

|                                |                             |
|--------------------------------|-----------------------------|
| (temp = buffReader.readLine()) | && buffReader.hasNext()     |
| (temp = buffReader.nextLine()) | (IOException e) {           |
| != null                        | (FileNotFoundException e) { |

Done

**Q: 50** Assuming that the serializeBanana() and the deserializeBanana() methods will correctly use Java serialization and given:

```
13. import java.io.*;
14. class Food implements Serializable {int good = 3;}
15. class Fruit extends Food {int juice = 5;}
16. public class Banana extends Fruit {
17.int yellow= 4;
18.public static void main(String [] args)
19.Banana b = new Banana(); Banana b2 = new Banana();
20.b.serializeBanana(b);// assume correct serialization
21.b2 = b.deserializeBanana(); // assume correct
22.System.out.println("restore "+b2.yellow+ b2.juice+b2.good);
24.}
25.// more Banana methods go here 50. }
```

What is the result?

- A. restore 400
- B. restore 403
- C. restore 453
- D. Compilation fails.
- E. An exception is thrown at runtime.

Answer: C

**Q: 51** Given this method in a class:

```
21. public String toString(){
22.StringBuffer buffer = new StringBuffer();
```

```

23.buffer.append('<');
24.buffer.append(this.name);
25.buffer.append('>');
26.return buffer.toString();
27.}

```

Which statement is true?

- A. This code is NOT thread-safe.
- B. The programmer can replace StringBuffer with StringBuilder with no other changes.
- C. This code will perform poorly. For better performance, the code should be rewritten:  

```
return "<" + this.name + ">"
```
- D. This code will perform well and converting the code to use StringBuilder will not enhance the performance.

Answer: B

Q: 52 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());
 }
}

```

Place the names "A" and "B" in the following output.

```

class A has name A
class B has name A

```

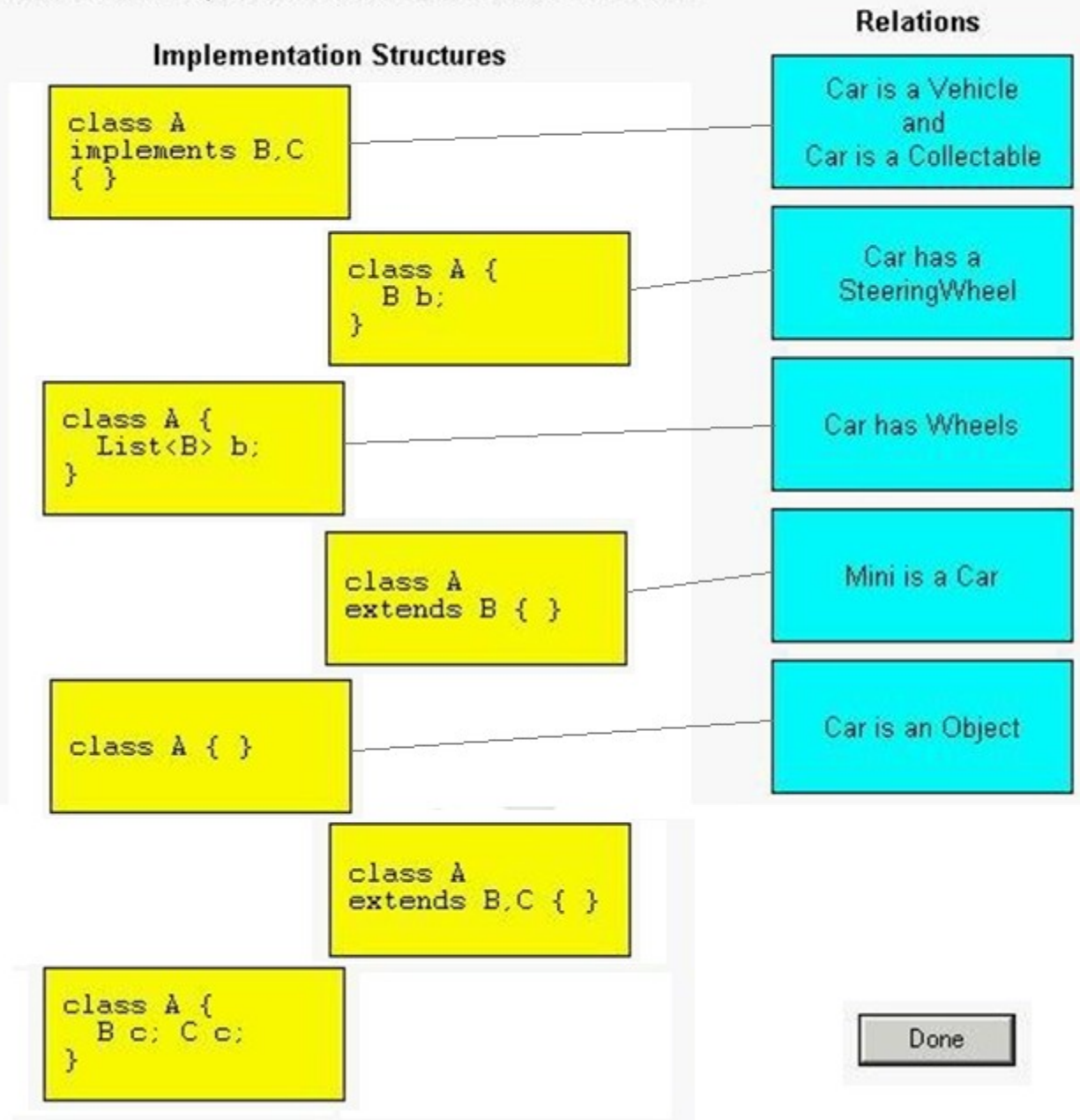
**Names**

A B

Done

Q: 53 Click the Task button.

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



Q: 54 Given:

1. package geometry;
2. public class Hypotenuse {
3. public InnerTriangle it= new InnerTriangle();
4. class InnerTriangle {
5. public int base;
6. public int height;
7. }
8. }

Which statement is true about the class of an object that can reference the variable base?

- A. It can be any class.
- B. No class has access to base.
- C. The class must belong to the geometry package.
- D. The class must be a subclass of the class Hypotenuse.

Answer: C

Q: 55 Which two statements are true about has-a and is-a relationships? (Choose two.)

- A. Inheritance represents an is-a relationship.
- B. Inheritance represents a has-a relationship.
- C. Interfaces must be used when creating a has-a relationship.
- D. Instance variables can be used when creating a has-a relationship.

Answer: A, D

Q: 56 Given:

```
10: public class Hello {
11: String title;
12: int value;
13: public Hello(){
14: title+= " World";
15: }
16: public Hello(int value){
17: this.value = value;
18: title= "Hello";
19: Hello();
20: }
21: }
and:
30: Hello c = new Hello(5);
31: System.out.println(c.title);
```

What is the result?

- A. Hello
- B. Hello World
- C. Compilation fails.
- D. Hello World 5
- E. The code runs with no output.
- F. An exception is thrown at runtime.

Answer: C

Q: 57 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); }

Answer: C, D

Q: 58 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.

Answer: A

Q: 59 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.

Answer: F

Q: 60 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?

- A. foofoofoofoofoo
- B. foobarfoobarbar
- C. foobarfoofoofoo
- D. foobarfoobarfoo
- E. barbarbarbarbar
- F. foofoofoobarbar
- G. foofoofoobarfoo

Answer: D

**Q: 61** 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.

Answer: A, D

**Q: 62** Given classes defined in two different files:

```
1. package util;
2. public class BitUtils{
3. public static void process(byte[]){/* more code here */ }
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. util.BitUtils.process(bytes);
- D. SomeApp cannot use methods in BitUtils.
- E. import util.BitUtils.\*; process(bytes);

Answer: C

**Q: 63** Given:

```
13. public class Pass {
14. public static void main(String [] args){
15. int x = 5;
16. Pass p = new Pass();
17. p.doStuff(x);
18. System.out.print(" main x = " + x);
19. }
20.
21. void doStuff(int x){
22. System.out.print(" doStuff x = " + x++);
23. }
24. }
```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. doStuff x = 6 main x = 6
- D. doStuff x = 5 main x = 5
- E. doStuff x = 5 main x = 6
- F. doStuff x = 6 main x = 5

Answer: D

Q: 64 Given:

```
11. public static void test(String str) {
12. if(str == null | str.length() == 0) {
13. System.out.println("String is empty");
14. } else{
15. System.out.println("String is not empty");
16. }
17. }
```

And the invocation:

```
31. test(null);
```

What is the result?

- A. An exception is thrown at runtime.
- B. "String is empty" is printed to output.
- C. Compilation fails because of an error in line 12.
- D. "String is not empty" is printed to output.

Answer: A

Q: 65 Given:

```
12. public class Yippee2 {
13.
14. static public void main(String [] yahoo){
15. for(int x = 1; x < yahoo.length; x++) {
16. System.out.print(yahoo[x] + " ");
17. }
18. }
19. }
```

and the command line invocation:

```
java Yippee2 a b c
```

What is the result?

- A. a b
- B. b c
- C. a b c
- D. Compilation fails.
- E. An exception is thrown at runtime.

Answer: B

Q: 66 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

Q: 67Click the Exhibit button.

```

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. }

```

Which three code fragments, added individually at line 29, produce the output 100? (Choose three.)

- A. n = 100;
- B. i.setX( 100 );

- C. o.getY().setX( 100 );
- D. i = new Inner(); i.setX( 100 );
- E. o.setY( i ); i = new Inner(); i.setX( 100 );
- F. i = new Inner(); i.setX( 100 ); o.setY( i );

Answer: B, C, F

Q: 68 Given:

```
15. public class Yippee {
16. public static void main(String [] args){
17. for(int x = 1; x < args.length; x++){
18. System.out.print(args[x]+ " ");
19. }
20. }
21. }
```

and two separate command line invocations:

```
java Yippee
java Yippee 1 2 3 4
```

What is the result?

- A. No output is produced.  
1 2 3
- B. No output is produced.  
2 3 4
- C. No output is produced.  
1 2 3 4
- D. An exception is thrown at runtime.  
1 2 3
- E. An exception is thrown at runtime.  
2 3 4
- F. An exception is thrown at runtime.  
1 2 3 4

Answer: B

Q: 69 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.

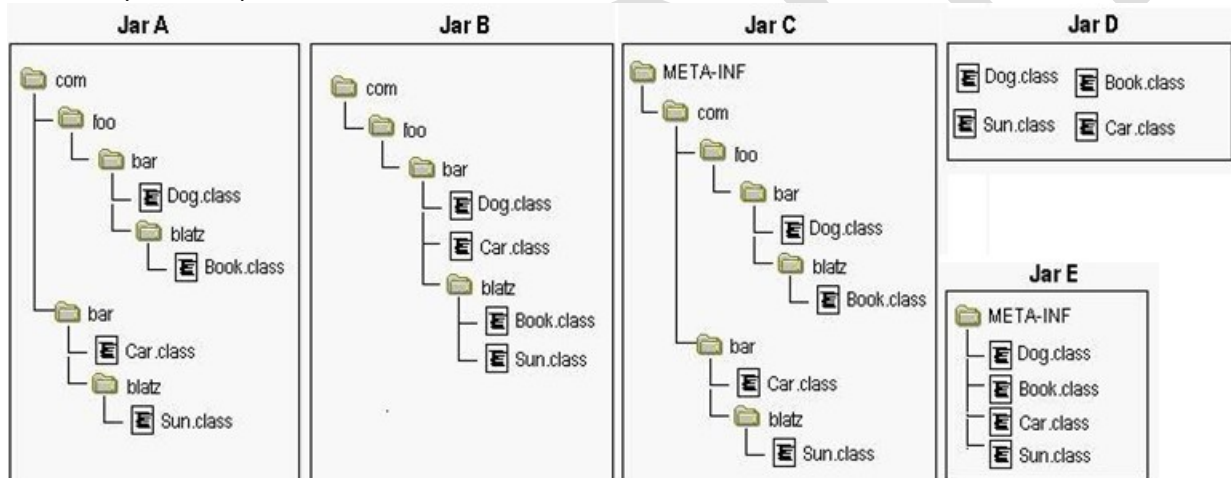
Answer: D

Q: 70 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

**Q: 71** 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

**Q: 72** Given a class Repetition:

```
1.package utils;
2.
3.public class Repetition{
4.public static String twice(String s){ return s + s;}
5.}
```

and given another class Demo:

```
1.// insert code here
2.
3.public class Demo {
4.public static void main(String[] args){
5.System.out.println(twice("pizza"));
6.}
7.}
```

Which code should be inserted at line 1 of Demo.java to compile and run Demo to print "pizzapizza"?

- A. import utils.\*;
- B. static import utils.\*;
- C. import utils.Repetition.\*;
- D. static import utils.Repetition.\*;
- E. import utils.Repetition.twice();
- F. import static utils.Repetition.twice;
- G. static import utils.Repetition.twice;

Answer: F

**Q: 73** 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

Q: 74 Given:

```
11. static class A{
12. void process() throws Exception { throw new Exception(); }
13. }
14. static class B extends A {
15. void process(){ System.out.println("B ");}
16. }
17. public static void main(String[] args){
18. A a = new B();
19. a.process();
20. }
```

What is the result?

- A. B
- B. The code runs with no output.
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error in line 15.
- E. Compilation fails because of an error in line 18.
- F. Compilation fails because of an error in line 19.

Answer: F

Q: 75 Given:

```
11. public void testIfA(){
12. if(testIfB("True")){
13. System.out.println("True");
14. } else{
15. System.out.println("Not true");
16. }
17. }
18. public Boolean testIfB(String str){
19. return Boolean.valueOf(str);
20. }
```

What is the result when method testIfA is invoked?

- A. True
- B. Not true
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error at line 12.
- E. Compilation fails because of an error at line 19.

Answer: A

Q: 76 Given:

```
1. public class Boxer1{
2.Integer i;
3.int x;
4.public Boxer1(int y) {
5.x = i+y;
6.System.out.println(x);
7.}
8.public static void main(String[] args){
9.new Boxer1(new Integer(4));
10.}
11.}
```

What is the result?

- A. The value "4" is printed at the command line.
- B. Compilation fails because of an error in line 5.
- C. Compilation fails because of an error in line 9.
- D. A NullPointerException occurs at runtime.
- E. A NumberFormatException occurs at runtime.
- F. An IllegalStateException occurs at runtime.

Answer: D

Q: 77 Given:

```
11. public static Iterator reverse(List list){
12.Collections.reverse(list);
13.return list.iterator();
14.}
15. public static void main(String[] args){
16.List list = new ArrayList();
17.list.add("1"); list.add("2"); list.add("3");
18.for (Object obj: reverse(list))
19.System.out.print(obj+ " ");
20.}
```

What is the result?

- A. 3, 2, 1,
- B. 1, 2, 3,
- C. Compilation fails.
- D. The code runs with no output.
- E. An exception is thrown at runtime.

Answer: C

Q: 78 Click the Task 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

Q: 79 Given:

```

23. int z = 5;
24.
25. public void stuff1(int x){
26. assert (x > 0);
27. switch(x){
28. case 2: x = 3;
29. default: assert false;}}
30.
31. private void stuff2(int y){ assert (y < 0); }
32.
33. private void stuff3(){ assert (stuff4());}

```

34.

35. `private boolean stuff4(){ z = 6; return false;}`

Which statement is true?

- A. All of the assert statements are used appropriately.
- B. Only the assert statement on line 31 is used appropriately.
- C. The assert statements on lines 29 and 31 are used appropriately.
- D. The assert statements on lines 26 and 29 are used appropriately.
- E. The assert statements on lines 29 and 33 are used appropriately.
- F. The assert statements on lines 29, 31, and 33 are used appropriately.
- G. The assert statements on lines 26, 29, and 31 are used appropriately.

Answer: C

Q: 80 Given:

25. `int x = 12;`

26. `while (x < 10){`

27. `x--;`

28. `}`

29. `System.out.print(x);`

What is the result?

- A. 0
- B. 10
- C. 12
- D. Line 29 will never be reached.

Answer: C

Q: 81 Given:

33. `try{`

34. `// some code here`

35. `} catch (NullPointerException e1) {`

36. `System.out.print("a");`

37. `} catch (RuntimeException e2) {`

38. `System.out.print("b");`

39. `} finally{`

40. `System.out.print("c");`

41. `}`

What is the result if a `NullPointerException` occurs on line 34?

- A. c
- B. a
- C. ab
- D. ac
- E. bc
- F. abc

Answer: D

**Q: 82** Given:

```
10. public class Foo {
11. static int[] a;
12. static { a[0]=2;}
13. public static void main(String[] args){
14. }
```

Which exception or error will be thrown when a programmer attempts to run this code?

- A. java.lang.StackOverflowError
- B. java.lang.IllegalStateException
- C. java.lang.ExceptionInInitializerError
- D. java.lang.ArrayIndexOutOfBoundsException

Answer: C

**Q: 83** Given:

```
11. public static void main(String[] args){
12. try{
13. args = null;
14. args[0] = "test";
15. System.out.println(args[0]);
16. } catch(Exception ex) {
17. System.out.println("Exception");
18. } catch(NullPointerException npe) {
19. System.out.println("NullPointerException");
20. }
21. }
```

What is the result?

- A. test
- B. Exception
- C. Compilation fails.
- D. NullPointerException

Answer: C

**Q: 84** 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?

- A. short LONG
- B. SHORT LONG
- C. Compilation fails.
- D. An exception is thrown at runtime.

Answer: C

Q: 85 Given:

```
12. import java.io.*;
13. public class Forest implements Serializable {
14. private Tree tree = new Tree();
15. public static void main(String [] args){
16. Forest f = new Forest();
17. try{
18. FileOutputStream fs = new FileOutputStream("Forest.ser");
19. ObjectOutputStream os = new ObjectOutputStream(fs);
20. os.writeObject(f); os.close();
21. } catch(Exception ex) { ex.printStackTrace();}
22. }}
23.
24. class Tree {}
```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. An instance of Forest is serialized.
- D. An instance of Forest and an instance of Tree are both serialized.

Answer: B

Q: 86 Given:

```
1. public class TestString3{
2. public static void main(String[] args){
3. // insert code here
5. System.out.println(s);
6. }
7. }
```

Which two code fragments, inserted independently at line 3, generate the output 4247? (Choose two.)

- A. String s = "123456789";  
s = (s-"123").replace(1,3,"24")- "89";
- B. StringBuffer s = new StringBuffer("123456789");  
s.delete(0,3).replace(1,3,"24").delete(4,6);
- C. StringBuffer s = new StringBuffer("123456789");  
s.substring(3,6).delete(1,3).insert(1, "24");
- D. StringBuilder s = new StringBuilder("123456789");  
s.substring(3,6).delete(1,2).insert(1, "24");



E. `StringBuilder s = new StringBuilder("123456789");`  
`s.delete(0,3).delete(1,3).delete(2,5).insert(1, "24");`

Answer: B, E

Q: 87 Click the Task button.

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

and E is approximately

Values

|                                |                                       |                                    |                                      |
|--------------------------------|---------------------------------------|------------------------------------|--------------------------------------|
| <input type="text" value="3"/> | <input type="text" value="3.141593"/> | <input type="text" value="true"/>  | <input type="text" value="Math.PI"/> |
| <input type="text" value="2"/> | <input type="text" value="2.718282"/> | <input type="text" value="false"/> | <input type="text" value="Math.E"/>  |

Q: 88 When comparing `java.io.BufferedWriter` to `java.io.FileWriter`, which capability exists as a method in only one of the two?

- A. closing the stream
- B. flushing the stream
- C. writing to the stream
- D. marking a location in the stream
- E. writing a line separator to the stream

Answer: E

Q: 89 Given:

```
12. Date date = new Date();
13. df.setLocale(Locale.ITALY);
14. String s = df.format(date);
```

The variable `df` is an object of type `DateFormat` that has been initialized in line 11.

What is the result if this code is run on December 14, 2000?

- A. The value of `s` is 14-dic-2004.
- B. The value of `s` is Dec 14, 2000.
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error in line 13.

Answer: D

Q: 90 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.

Answer: B, D, E

Q: 91 Click the Task button.

The `doesFileExist` method takes an array of directory names representing a path from the root filesystem and a file name. The method returns true if the file exists, false if it does not.

Place the code fragments in position to complete this method.

```
public static boolean doesFileExist(String[] directories, String filename) {
 String path = "";
 for (String dir : directories) {
 path = path + File.separator + dir;
 }
 File file = new File(path, filename);
 return file.exists();
}
```

#### Code Fragments

|                                                |                                                    |                                                    |
|------------------------------------------------|----------------------------------------------------|----------------------------------------------------|
| <code>path = path.getSubdirectory(dir);</code> | <code>return ! file.isNew();</code>                | <code>return (file != null);</code>                |
| <code>String path = "";</code>                 | <code>path = path.getFile(filename);</code>        | <code>File path = new File("");</code>             |
| <code>return file.exists();</code>             | <code>return path.isFile();</code>                 | <code>File file = new File(path, filename);</code> |
| <code>path = new File(path, dir);</code>       | <code>File path = new File(File.separator);</code> | <code>path = path + File.separator + dir;</code>   |

Q: 92 Click the Exhibit button.

Which code, inserted at line 14, will allow this class to correctly serialize and deserialize?

```

1. import java.io.*;
2. public class Foo implements Serializable{
3. public int x, y;
4. public Foo(int x, int y) { this.x = x; this.y = y; }
5.
6. private void writeObject(ObjectOutputStream s)
7. throws IOException {
8. s.writeInt(x); s.writeInt(y) ;
9. }
10.
11. private void readObject(ObjectInputStream s)
12. throws IOException, ClassNotFoundException {
13.
14. // insert code here
15.
16. }
17. }

```

- A. s.defaultReadObject();
- B. this = s.defaultReadObject();
- C. y = s.readInt(); x = s.readInt();
- D. x = s.readInt(); y = s.readInt();

Answer: D

Q: 93 Given:

```

11. String test = "This is a test";
12. String[] tokens = test.split("\s");
13. System.out.println(tokens.length);

```

What is the result?

- A. 0
- B. 1
- C. 4
- D. Compilation fails.
- E. An exception is thrown at runtime.

Answer: D

**Q: 94 Given:**

d is a valid, non-null Date object

df is a valid, non-null DateFormat object set to the current locale

What outputs the current locale's country name and the appropriate version of d's date?

- A. Locale loc = Locale.getLocale();  
System.out.println(loc.getDisplayCountry()  
+ " " + df.format(d));
- B. Locale loc = Locale.getDefault();  
System.out.println(loc.getDisplayCountry()  
+ " " + df.format(d));
- C. Locale loc = Locale.getLocale();

```

 System.out.println(loc.getDisplayCountry()
 + " " + df.setDateFormat(d));
D. Locale loc = Locale.getDefault();
 System.out.println(loc.getDisplayCountry()
 + " " + df.setDateFormat(d));

```

Answer: B

Q: 95 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: 96 Given:

```

11. rbo = new ReallyBigObject();
12. // more code here
13. rbo = null;
14. /* insert code here */

```

Which statement should be placed at line 14 to suggest that the virtual machine expend effort toward recycling the memory used by the object rbo?

- A. System.gc();
- B. Runtime.gc();
- C. System.freeMemory();
- D. Runtime.getRuntime().growHeap();
- E. Runtime.getRuntime().freeMemory();

Answer: A

Q: 97 Given classes defined in two different files:

```

1. package util;
2. public class BitUtils{
3. public static void process(byte[]){/* more code here */ }
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. util.BitUtils.process(bytes);
- D. SomeApp cannot use methods in BitUtils.
- E. import util.BitUtils.\*; process(bytes);

Answer: C

**Q: 98 Given:**

```

11. public static void test(String str) {
12. int check = 4;
13. if(check == str.length()){
14. System.out.print(str.charAt(check - 1)+", ");
15. } else{
16. System.out.print(str.charAt(0)+ ", ");
17. }
18. }

```

and the invocation:

```

21. test("four");
22. test("tee");
23. test("to");

```

What is the result?

- A. r, t, t,
- B. r, e, o,
- C. Compilation fails.
- D. An exception is thrown at runtime.

Answer: C

**Q: 99 Given:**

```

11. public class Commander {
12. public static void main(String[] args){
13. String myProp = /* insert code here */
14. System.out.println(myProp);

```

15.}

16.}

and the command line:

java -Dprop.custom=gobstopper Commander

Which two, placed on line 13, will produce the output gobstopper? (Choose two.)

- A. System.load("prop.custom");
- B. System.getenv("prop.custom");
- C. System.property("prop.custom");
- D. System.getProperty("prop.custom");
- E. System.getProperties().getProperty("prop.custom");

Answer: D, E

**Q: 100 Given:**

```
11. class Snoochy {
12.Boochy booch;
13.public Snoochy() { booch = new Boochy(this); }
14.}
```

15.

```
16. class Boochy {
17.Snoochy snooch;
18.public Boochy(Snoochy s) { snooch = s; }
19.}
```

And the statements:

```
21. public static void main(String[] args){
22.Snoochy snoog = new Snoochy();
23.snoog = null;
24.// more code here
25.}
```

Which statement is true about the objects referenced by snoog, snooch, and booch immediately after line 23 executes?

- A. None of these objects are eligible for garbage collection.
- B. Only the object referenced by booch is eligible for garbage collection.
- C. Only the object referenced by snoog is eligible for garbage collection.
- D. Only the object referenced by snooch is eligible for garbage collection.
- E. The objects referenced by snooch and booch are eligible for garbage collection.

Answer: E

**Q: 101** Click the Exhibit button.

What is the outcome of the code?

```

1. public class Item {
2. private String desc;
3. public String getDescription() { return desc; }
4. public void setDescription(String d) { desc = d; }
5.
6. public static void modifyDesc(Item item, String desc) {
7. item = new Item();
8. item.setDescription(desc);
9. }
10. public static void main(String[] args){
11. Item it = new Item();
12. it.setDescription("Gobstopper");
13. Item it2 = new Item();
14. it2.setDescription("Fizzylifting");
15. modifyDesc(it,"Scrumdiddlyumptious");
16. System.out.println(it.getDescription());
17. System.out.println(it2.getDescription());
18. }
19. }

```

- A. Compilation fails.
- B. Gobstopper Fizzylifting
- C. Gobstopper Scrumdiddlyumptious
- D. Scrumdiddlyumptious Fizzylifting
- E. Scrumdiddlyumptious

Answer: B

Q: 102 Given:

```

12. public class Yippee2 {
13.
14. static public void main(String [] yahoo){
15. for(int x = 1; x < yahoo.length; x++) {
16. System.out.print(yahoo[x] + " ");
17. }
18. }
19. }

```

and the command line invocation:

java Yippee2 a b c

What is the result?

- A. a b
- B. b c
- C. a b c
- D. Compilation fails.
- E. An exception is thrown at runtime.

Answer: B

Q: 103 Click the Exhibit button.

What is the output of the program shown in the exhibit?

```
10. class Foo {
11. private int x;
12. public Foo(int x) { this.x = x; }
13. public void setX(int x) { this.x = x; }
14. public int getX() { return x; }
15. }
16.
17. public class Gamma {
18.
19. static Foo fooBar(Foo foo) {
20. foo = new Foo(100);
21. return foo;
22. }
23.
24. public static void main(String[] args
25.) {
26. Foo foo = new Foo(300);
27. System.out.print(foo.getX() + "-");
28.
29. Foo fooFoo = fooBar(foo);
30. System.out.print(foo.getX() + "-");
31. System.out.print(fooFoo.getX() + "-");
32.
33. foo = fooBar(fooFoo);
34. System.out.print(foo.getX() + "-");
35. System.out.print(fooFoo.getX());
36. }
37. }
```

- A. 300-100-100-100-100
- B. 300-300-100-100-100
- C. 300-300-300-100-100
- D. 300-300-300-300-100

Answer: B

Q: 104 A developer is creating a class Book, that needs to access class Paper. The Paper class is deployed in a JAR named myLib.jar. Which three, taken independently, will allow the developer to use the Paper class while compiling the Book class? (Choose three.)

- A. The JAR file is located at \$JAVA\_HOME/jre/classes/myLib.jar.
- B. The JAR file is located at \$JAVA\_HOME/jre/lib/ext/myLib.jar.
- C. The JAR file is located at /foo/myLib.jar and a classpath environment variable is set that includes /foo/myLib.jar/Paper.class.
- D. The JAR file is located at /foo/myLib.jar and a classpath environment variable is set that includes /foo/myLib.jar
- E. The JAR file is located at /foo/myLib.jar and the Book class is compiled using javac -cp /foo/myLib.jar/PaperBook.java.
- F. The JAR file is located at /foo/myLib.jar and the Book class is compiled using javac -d /foo/myLib.jar Book.java



G. The JAR file is located at /foo/myLib.jar and the Book class is compiled using `javac -classpath /foo/myLib.jar Book.java`

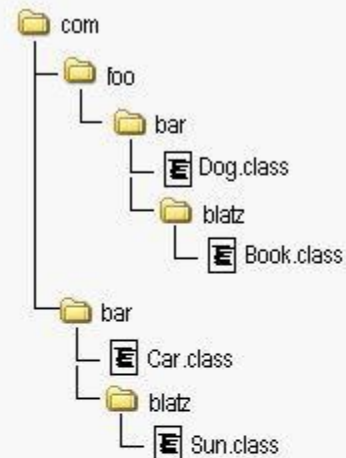
**Answer: B, D, G**

**Q: 105 Click the Task button.**

The image at right represents a complete package structure for a set of classes: "com" is the beginning of the fully-qualified package name for all classes.

Given this package structure, insert the code needed to make the Car class compile and run successfully.

All three placeholders must be filled. If fewer than three statements are needed, use the "// blank" option.



`package com.bar;`

`import com.foo.bar.*;`

`import com.foo.bar.blatz.*;`

```
public class Car {
 Book book;
 Dog dog;
}
```

`import com.foo.bar.blatz.*;`

`package com.foo.bar.blatz;`

`import com.bar.*;`

`import com.*;`

`package com.bar;`

`package com;`

`import com.foo.*;`

`// blank`

`import com.foo.bar.*;`

`import com.foo.bar.Book;`

Done

**Q: 106 Click the Task button.**

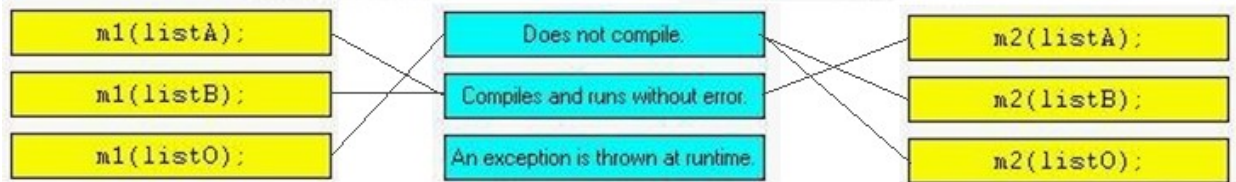
Given:

```
1. import java.util.*;
2. class A { }
3. class B extends A { }
4. public class Test {
5. public static void main(String[] args) {
6. List<A> listA = new LinkedList<A>();
7. List listB = new LinkedList();
8. List<Object> listO = new LinkedList<Object>();
9. // insert code here
10. }
11. public static void m1(List<? extends A> list) { }
12. public static void m2(List<A> list) { }
13. }
```

Place a result onto each method call to indicate what would happen if the method call were inserted at line 9. Note: Results can be used more than once.

| Method Calls            |                         | Result                             |
|-------------------------|-------------------------|------------------------------------|
| <code>m1(listA);</code> | <code>m2(listA);</code> | Does not compile.                  |
| <code>m1(listB);</code> | <code>m2(listB);</code> | Compiles and runs without error.   |
| <code>m1(listO);</code> | <code>m2(listO);</code> | An exception is thrown at runtime. |

Answer:



Q: 107 Given:

```
enum Example { ONE, TWO, THREE }
```

Which statement is true?

- A. The expressions `(ONE == ONE)` and `ONE.equals(ONE)` are both guaranteed to be true.
- B. The expression `(ONE < TWO)` is guaranteed to be true and `ONE.compareTo(TWO)` is guaranteed to be less than one.
- C. The Example values cannot be used in a raw `java.util.HashMap`; instead, the programmer must use a `java.util.EnumMap`.
- D. The Example values can be used in a `java.util.SortedSet`, but the set will NOT be sorted because enumerated types do NOT implement `java.lang.Comparable`.

Answer: A

Q: 108 Given:

```

1. import java.util.*;
2. public class PQ {
3. public static void main(String[] args){
4. PriorityQueue<String> pq = new PriorityQueue<String>();
5. pq.add("carrot");
6. pq.add("apple");
7. pq.add("banana");
8. System.out.println(pq.poll() + ":" + pq.peek());
9. }
10. }

```

What is the result?

- A. apple:apple
- B. carrot:apple
- C. apple:banana
- D. banana:apple
- E. carrot:carrot

F. carrot:banana

Answer: C

Q: 110 Given:

```
1. import java.util.*;
2. public class WrappedString {
3. private String s;
4. public WrappedString(String s) { this.s = s; }
5. public static void main(String[] args){
6. HashSet<Object> hs = new HashSet<Object>();
7. WrappedString ws1 = new WrappedString("aardvark");
8. WrappedString ws2 = new WrappedString("aardvark");
9. String s1 = new String("aardvark");
10. String s2 = new String("aardvark");
11. hs.add(ws1); hs.add(ws2); hs.add(s1); hs.add(s2);
12. System.out.println(hs.size());}}
```

What is the result?

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4
- F. Compilation fails.
- G. An exception is thrown at runtime.

Answer: D

Q: 111 Given:

```
11. public class Key {
12. private long id1;
13. private long id2;
14.
15. // class Key methods
16. }
```

A programmer is developing a class Key, that will be used as a key in a standard java.util.HashMap. Which two methods should be overridden to assure that Key works correctly as a key? (Choose two.)

- A. public int hashCode()
- B. public boolean equals(Key k)
- C. public int compareTo(Object o)
- D. public boolean equals(Object o)
- E. public boolean compareTo(Key k)

Answer: A, D

Q: 112 Given a pre-generics implementation of a method:

```

11. public static int sum(List list){
12.int sum = 0;
13.for (Iterator iter = list.iterator(); iter.hasNext());{
14.int i= ((Integer)iter.next()).intValue();
15.sum += i;
16.}
17.return sum;
18.}

```

Which three changes must be made to the method sum to use generics? (Choose three.)

- A. remove line 14
- B. replace line 14 with "int i = iter.next();"
- C. replace line 13 with "for (int i : intList) {"
- D. replace line 13 with "for (Iterator iter : intList) {"
- E. replace the method declaration with "sum(List<int> intList)"
- F. replace the method declaration with "sum(List<Integer> intList)"

Answer: A, C, F

**Q: 113** Click the Task button.

Given: `NumberNames nn = new NumberNames();`  
`nn.put("one", 1);`  
`System.out.println(nn.getNames());`

Place the code into position to create a class that maps from Strings to integer values. The result of execution must be [one]. Some options may be used more than once.

```

public class NumberNames {
 private HashMap<String, Integer> map =
 new HashMap<String, Integer> >();
 public void put(String name, int value) {
 map.put(name, value);
 }
 public Set<String> getNames() {
 return map.keySet();
 }
}

```

**Code**

|                      |                  |                      |     |
|----------------------|------------------|----------------------|-----|
| Set<int>             | Set<Integer>     | HashSet              |     |
| Set<Integer, String> | Set<int, String> | Set<String, Integer> |     |
| Set<String, int>     | Set<String>      | NumberNames          |     |
| String               | Integer          | int                  | >   |
| >()                  | name             | value                | map |

**Q: 114 Given:**

11.// insert code here

```

12.private N min, max;
13.public N getMin() { return min; }
14.public N getMax() { return max; }
15.public void add(N added) {
16.if(min == null || added.doubleValue() < min.doubleValue())
17.min = added;
18.if(max == null || added.doubleValue() > max.doubleValue())
19.max = added;
20.}
21.}

```

Which two, inserted at line 11, will allow the code to compile? (Choose two.)

- A. public class MinMax<?> {
- B. public class MinMax<? extends Number> {
- C. public class MinMax<N extends Object> {
- D. public class MinMax<N extends Number> {
- E. public class MinMax<? extends Object> {
- F. public class MinMax<N extends Integer> {

Answer: D, F

**Q: 115 Given:**

```

1. import java.util.*;
2.
3. public class LetterASort{
4.public static void main(String[] args){
5.ArrayList<String> strings = new ArrayList<String>();
6.strings.add("aAaA");
7.strings.add("AaA");
8.strings.add("aAa");
9.strings.add("AAaa");
10.Collections.sort(strings);
11.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

**Q: 116 Given:**

```

11. public abstract class Shape {
12. private int x;
13. private int y;
14. public abstract void draw();
15. public void setAnchor(int x, int y){
16. this.x= x;
17. this.y= y;
18. }
19. }

```

Which two classes use the Shape class correctly? (Choose two.)

- A. public class Circle implements Shape { private int radius; }
- B. public abstract class Circle extends Shape {private int radius;}
- C. public class Circle extends Shape {private int radius; public void draw(); }
- D. public abstract class Circle extends Shape {private int radius; public void draw(); }
- E. public class Circle extends Shape {private int radius; public void draw(){/\* code here \*/} }
- F. public abstract class Circle implements Shape {private int radius; public void draw(){ /\* code here \*/} }

Answer: B, E

Q: 117 Given:

```

10. class Foo {
11. static void alpha(){/* more code here */ }
12. void beta(){/* more code here */ }
13. }

```

Which two statements are true? (Choose two.)

- A. Foo.beta() is a valid invocation of beta().
- B. Foo.alpha() is a valid invocation of alpha().
- C. Method beta() can directly call method alpha().
- D. Method alpha() can directly call method beta().

Answer: B, C

Q: 118 Given:

```

10. class One {
11. public One() { System.out.print(1);}
12. }
13. class Two extends One {
14. public Two() { System.out.print(2); }
15. }
16. class Three extends Two {
17. public Three() { System.out.print(3);}
18. }

```

```

19. public class Numbers{
20. public static void main(String[] argv){ new Three(); }
21. }

```

What is the result when this code is executed?

- A. 1
- B. 3
- C. 123
- D. 321
- E. The code runs with no output.

Answer: C

Q: 119 Click the Exhibit button.

Which three statements are true? (Choose three.)

```

10. interface Foo {
11. int bar();
12. }
13.
14. public class Beta {
15.
16. class A implements Foo {
17. public int bar() { return 1; }
18. }
19.
20. public int fubar(Foo foo) { return
foo.bar(); }
21.
22. public void testFoo() {
23.
24. class A implements Foo {
25. public int bar() { return 2; }
26. }
27.
28. System.out.println(fubar(new A())
);
29. }
30.
31. public static void main(String[] argv
) {
32. new Beta().testFoo();
33. }
34. }

```

- A. Compilation fails.
- B. The code compiles and the output is 2.
- C. If lines 16, 17 and 18 were removed, compilation would fail.
- D. If lines 24, 25 and 26 were removed, compilation would fail.
- E. If lines 16, 17 and 18 were removed, the code would compile and the output would be 2.
- F. If lines 24, 25 and 26 were removed, the code would compile and the output would be 1.

Answer: B, E, F

Q: 120 Click the Exhibit button.

Given:

34. Test t = new Test();

35. t.method(5);

What is the output from line 5 of the Test class?

```
1. public class Test {
2. int x = 12;
3. public void method(int x) {
4. x+=x;
5. System.out.println(x);
6. }
7. }
```

- A. 5
- B. 10
- C. 12
- D. 17
- E. 24

Answer: B

Q: 121 Given:

10. class Nav{

11. public enum Direction { NORTH, SOUTH, EAST, WEST }

12. }

13. public class Sprite{

14. // insert code here

15. }

Which code, inserted at line 14, allows the Sprite class to compile?

- A. Direction d = NORTH;
- B. Nav.Direction d = NORTH;
- C. Direction d = Direction.NORTH;
- D. Nav.Direction d = Nav.Direction.NORTH;

Answer: D

Q: 122 Given:

10. class One {

11. public One foo() { return this; }

12. }

13. class Two extends One {

14. public One foo() { return this; }

15. }

16. class Three extends Two {

17. // insert method here

18. }



Which two methods, inserted individually, correctly complete the Three class? (Choose two.)

- A. `public void foo(){}`
- B. `public int foo() { return 3; }`
- C. `public Two foo() { return this; }`
- D. `public One foo() { return this; }`
- E. `public Object foo() { return this; }`

Answer: C, D

Q: 123 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 {
```

```
 public void bar(int x) {}
```

```
 public int bar(String x) { return 1; }
```

```
 public void bar(int x, int y) {}
```

```
}
```

#### 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; }
```

Q: 125 Click the Exhibit button.

```
11. public class Bootchy {
12. int bootch;
13. String snootch;
14.
15. public Bootchy() {
16. this("snootchy");
17. System.out.print("first ");
18. }
19.
20. public Bootchy(String snootch) {
21. this(420, "snootchy");
22. System.out.print("second ");
23. }
24.
25. public Bootchy(int bootch, String
snootch) {
26. this.bootch = bootch;
27. this.snootch = snootch;
28. System.out.print("third ");
29. }
30.
```

What is the result?

- A. snootchy 420 third second first

- B. snootchy 420 first second third
- C. first second third snootchy 420
- D. third second first snootchy 420
- E. third first second snootchy 420
- F. first second first third snootchy 420

Answer: D

Q: 126 A JavaBeans component has the following field:

11. private boolean enabled;

Which two pairs of method declarations follow the JavaBeans standard for accessing this field? (Choose two.)

- A. public void setEnabled( boolean enabled )  
public boolean getEnabled()
- B. public void setEnabled( boolean enabled )  
public void isEnabled()
- C. public void setEnabled( boolean enabled )  
public boolean isEnabled()
- D. public boolean setEnabled( boolean enabled )  
public boolean getEnabled()

Answer: A, C

Q: 127 Given:

```
11. public static void main(String[] args){
12. Object obj = new int[] { 1, 2, 3 };
13. int[] someArray = (int[])obj;
14. for (int i : someArray) System.out.print(i + " ");
15. }
```

What is the result?

- A. 1 2 3
- B. Compilation fails because of an error in line 12.
- C. Compilation fails because of an error in line 13.
- D. Compilation fails because of an error in line 14.
- E. A ClassCastException is thrown at runtime.

Answer: A

Q: 128 Given:

```
1. public class Threads3 implements Runnable {
2. public void run() {
3. System.out.print("running");
4. }
5. public static void main(String[] args){
6. Thread t = new Thread(new Threads3());
7. t.run();
```

```

8.t.run();
9.t.start();
10.}
11.}

```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes and prints "running".
- D. The code executes and prints "runningrunning".
- E. The code executes and prints "runningrunningrunning".

Answer: E

Q: 129 Click the Exhibit button.

Which two are possible results? (Choose two.)

```

1. public class Threadsl {
2. int x = 0;
3. public class Runner implements Runnable
4. {
5. public void run() {
6. int current = 0;
7. for(int i = 0; i < 4; i++) {
8. current = x;
9. System.out.print(current + ", ");
10. x = current + 2;
11. }
12. }
13. }
14. public static void main(String[] args) {
15. new Threadsl().go();
16. }
17.
18. public void go() {
19. Runnable r1 = new Runner();
20. new Thread(r1).start();
21. new Thread(r1).start();
22. }
23.}

```

- A. 0, 2, 4, 4, 6, 8, 10, 6,
- B. 0, 2, 4, 6, 8, 10, 2, 4,
- C. 0, 2, 4, 6, 8, 10, 12, 14,
- D. 0, 0, 2, 2, 4, 4, 6, 6, 8, 8, 10, 10, 12, 12, 14, 14,
- E. 0, 2, 4, 6, 8, 10, 12, 14, 0, 2, 4, 6, 8, 10, 12, 14,

Answer: A, C

Q: 130 Which three will compile and run without exception? (Choose three.)

- A. `private synchronized Object o;`
- B. `void go() {synchronized() {/* code here */}}`
- C. `public synchronized void go(){ /* code here */}`
- D. `private synchronized(this) void go(){ /* code here */}`
- E. `void go() {synchronized(Object.class){ /* code here */}}`
- F. `void go() {Object o= new Object();  
synchronized(o){ /* code here */}}`

Answer: C, E, F

**Q: 131 Given:**

```
1. public class TestOne {
2. public static void main (String[] args) throws Exception {
3. Thread.sleep(3000);
4. System.out.println("sleep");
5. }
6. }
```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes normally and prints "sleep".
- D. The code executes normally, but nothing is printed.

Answer: C

**Q: 132 Given:**

```
7. void waitForSignal() {
8. Object obj = new Object();
9. synchronized (Thread.currentThread()) {
10. obj.wait();
11. obj.notify();
12. }
13. }
```

Which statement is true?

- A. This code may throw an `InterruptedException`.
- B. This code may throw an `IllegalStateException`.
- C. This code may throw a `TimeoutException` after ten minutes.
- D. This code will not compile unless `obj.wait()` is replaced with `((Thread) obj).wait()`.
- E. Reversing the order of `obj.wait()` and `obj.notify()` may cause this method to complete normally.
- F. A call to `notify()` or `notifyAll()` from another thread may cause this method to complete normally.

Answer: B

**Q: 133 Click the Exhibit button.**

What is the result?

```

1. class Computation extends Thread {
2.
3. private int num;
4. private boolean isComplete;
5. private int result;
6.
7. public Computation(int num) { this.num = num; }
8.
9. public synchronized void run() {
10. result = num * 2;
11. isComplete = true;
12. notify();
13. }
14.
15. public synchronized int getResult() {
16. while (!isComplete) {
17. try {
18. wait();
19. } catch (InterruptedException e){}
20. }
21. return result;
22. }
23.
24. public static void main(String[] args){
25. Computation[] computations = new Computation[4];
26. for (int i = 0; i < computations.length; i++) {
27. computations[i] = new Computation(i);
28. computations[i].start();
29. }
30. for (Computation c : computations)
31. System.out.print(c.getResult() + " ");
32. }
33. }

```

- A. The code will deadlock.
- B. The code may run with no output.
- C. An exception is thrown at runtime.
- D. The code may run with output "0 6".
- E. The code may run with output "2 0 6 4".
- F. The code may run with output "0 2 4 6".

Answer: F

Q: 136 Click the Exhibit button.

What is the result?

```

1. public class SimpleCalc {
2. public int value;
3. public void calculate() { value += 7; }
4. }
And:
1. public class MultiCalc extends SimpleCalc{
2. public void calculate() { value -= 3; }
3. public void calculate(int multiplier) {
4. calculate();
5. super.calculate();
6. value *= multiplier;
7. }
8. public static void main(String[] args){
9. MultiCalc calculator = new MultiCalc();
10. calculator.calculate(2);
11. System.out.println("Value is: " + calculator.value)
12. }
13. }

```

- A. Value is: 8
- B. Compilation fails
- C. Value is: 12
- D. Value is: -12
- E. The code runs with no output.
- F. An exception is thrown at runtime.

Answer: A

Q: 137 Given:

```

10. interface A { public int getValue(); }
11. class B implements A {
12. public int getValue(){ return 1; }
13. }
14. class C extends B {
15. // insert code here
16. }

```

Which three code fragments, inserted individually at line 15, make use of polymorphism? (Choose three.)

- A. `public void add(C c) { c.getValue(); }`
- B. `public void add(B b) { b.getValue(); }`
- C. `public void add(A a) { a.getValue(); }`
- D. `public void add(A a, B b) { a.getValue(); }`
- E. `public void add(C c1, C c2) { c1.getValue(); }`

Answer: B, C, D

Q: 138 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.

Answer: C

Q: 139 Click the Task button.

Place the Types in one of the Type columns, and the Relationships in the Relationship column, to define appropriate has-a and is-a relationships.

| Type      | Relationship | Type             | Relationships | Types     |
|-----------|--------------|------------------|---------------|-----------|
| Dog       | is-a         | Animal           | is-a          | Dog       |
| Forest    | has-a        | Tree             | has-a         | Side      |
| Rectangle | has-a        | Side             |               | Tail      |
| Java Book | is-a         | Programming Book |               | Square    |
|           |              |                  |               | Tree      |
|           |              |                  |               | Book      |
|           |              |                  |               | Java Book |
|           |              |                  |               | Pen       |

Done

Q: 140 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: 141 Given:

```
11. class Animal { public String noise(){ return "peep"; } }
12. class Dog extends Animal {
13. public String noise(){ return "bark"; }
14. }
15. class Cat extends Animal {
16. public String noise(){ return "meow"; }
17. }
```

```
30. Animal animal = new Dog();
31. Cat cat = (Cat)animal;
32. System.out.println(cat.noise());
```

What is the result?

- A. peep
- B. bark
- C. meow
- D. Compilation fails.
- E. An exception is thrown at runtime.

Answer: E

Q: 142 Which three statements are true? (Choose three.)

- A. A final method in class X can be abstract if and only if X is abstract.
- B. A protected method in class X can be overridden by any subclass of X.
- C. A private static method can be called only within other static methods in class X.
- D. A non-static public final method in class X can be overridden in any subclass of X.
- E. A public static method in class X can be called by a subclass of X without explicitly referencing the class X.
- F. A method with the same signature as a private final method in class X can be implemented in a subclass of X.
- G. 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.

Answer: B, E, F

Q: 143 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 final Single getInstance() {
 if (instance == null) instance = create();
 return instance;
 }
 protected Single() { }
 protected Single create() { return new Single(); }
}

class SingleSub extends Single {
}
```

#### Modifiers

final

protected

private

abstract

static

Done

Q: 144 Click the Task 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;`

Answer: D, E

Q: 145 Which two statements are true about has-a and is-a relationships? (Choose two.)

- A. Inheritance represents an is-a relationship.
- B. Inheritance represents a has-a relationship.
- C. Interfaces must be used when creating a has-a relationship.
- D. Instance variables can be used when creating a has-a relationship.

Answer: A, D

Q: 146 Given:

```
1. public class TestOne {
2. public static void main (String[] args) throws Exception {
3. Thread.sleep(3000);
4. System.out.println("sleep");
5. }
6. } What is the result?
```

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes normally and prints "sleep".
- D. The code executes normally, but nothing is printed.

Answer: C

Q: 147 Click the Task button.

Given:

```
10. Runnable r = new Runnable() {
11. public void run() {
12. try {
13. Thread.sleep(1000);
14. } catch (InterruptedException e) {
15. System.out.println("interrupted");
16. }
17. System.out.println("ran");
18. }
19. };
20. Thread t = new Thread(r);
21. t.start();
22. System.out.println("started");
23. t.sleep(2000);
24. System.out.println("interrupting");
25. t.interrupt();
26. 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**

|                  |
|------------------|
| started          |
| ran              |
| interrupting     |
| ended            |
| (no more output) |

**Fragments**

|                       |
|-----------------------|
| interrupted           |
| ran                   |
| started               |
| interrupting          |
| ended                 |
| InterruptedException: |
| (no more output)      |

Done

Q: 148 Click the Task button.

What is the output if the main() method is run?

Given:

```
10. public class Starter extends Thread {
11. private int x = 2;
12. public static void main(String[] args)
throws Exception {
13. new Starter().makeItSo();
14. }
15. public Starter() {
16. x = 5;
17. start();
18. }
19. public void makeItSo() throws
Exception {
20. join();
21. x = x - 1;
22. System.out.println(x);
23. }
24. public void run() { x *= 2; }
25. }
```

- A. 4
- B. 5
- C. 8
- D. 9
- E. Compilation fails.
- F. An exception is thrown at runtime.
- G. It is impossible to determine for certain.

Answer: D

Q: 149 Given:

```
11. Runnable r = new Runnable() {
12. public void run() {
13. System.out.print("Cat");
14. }
15. };
16. Thread t = new Thread(r) {
17. public void run() {
18. System.out.print("Dog");
19. }
20. };
21. t.start();
```

What is the result?

- A. Cat
- B. Dog
- C. Compilation fails.

- D. The code runs with no output.
- E. An exception is thrown at runtime.

Answer: B

Q: 150 Given:

```
1. public class TestFive {
2. private int x;
3. public void foo(){
4. int current= x;
5. x = current + 1;
6. }
7. public void go() {
8. for(int i = 0; i < 5; i++) {
9. new Thread() {
10. public void run() {
11. foo();
12. System.out.print(x + ", ");
13. } }.start();
14. } }
```

Which two changes, taken together, would guarantee the output: 1, 2, 3, 4, 5, ? (Choose two.)

- A. move the line 12 print statement into the foo() method
- B. change line 7 to public synchronized void go() {
- C. change the variable declaration on line 2 to private volatile int x;
- D. wrap the code inside the foo() method with a synchronized( this ) block
- E. wrap the for loop code inside the go() method with a synchronized block synchronized(this){  
// for loop code here }

Q: 151 Click the Task button.

Place a Class on each method that is declared in the class.

| Class            | Method Name |                  |
|------------------|-------------|------------------|
| java.lang.Object | run()       | java.lang.Thread |
| java.lang.Thread | wait()      | java.lang.Object |
|                  | notify()    | java.lang.Object |
|                  | sleep()     | java.lang.Thread |
|                  | start()     | java.lang.Thread |
|                  | join()      | java.lang.Thread |

Done

Q: 152 Given:

foo and bar are public references available to many other threads. foo refers to a Thread and bar is an Object. The thread foo is currently executing bar.wait().

From another thread, what provides the most reliable way to ensure that foo will stop executing wait()?

- A. foo.notify();
- B. bar.notify();
- C. foo.notifyAll();
- D. Thread.notify();
- E. bar.notifyAll();
- F. Object.notify();

Answer: E

Q: 153 Click the Exhibit button.

```
1. import java.util.*;
2.
3. public class NameList {
4. private List names = new ArrayList();
5. public synchronized void add(String name) { names.add(name); }
6. public synchronized void printAll() {
7. for (int i = 0; i < names.size(); i++) {
8. System.out.print(names.get(i) + " ");
9. }
10. }
11. public static void main(String[] args){
12. final NameList sl = new NameList();
13. for (int i = 0; i < 2; i++) {
14. new Thread() {
15. public void run() {
16. sl.add("A");
17. sl.add("B");
18. sl.add("C");
19. sl.printAll();
20. }
21. }.start();
22. }
23. }
24. }
```

Which two statements are true if this class is compiled and run?(Choose two.)

- A. An exception may be thrown at runtime.
- B. The code may run with no output, without exiting.
- C. The code may run with no output, exiting normally.
- D. The code may run with output "A B A B C C ", then exit.
- E. The code may run with output "A B C A B C A B C ", then exit.
- F. The code may run with output "A A A B C A B C C ", then exit.
- G. The code may run with output "A B C A A B C A B C ", then exit.

Answer: E, G

Q: 154 Given:

```
11. static void test() throws RuntimeException {
12. try {
13. System.out.print("test ");
14. throw new RuntimeException();
15. }
16. catch (Exception ex) { System.out.print("exception "); }
17. }
18. public static void main(String[] args) {
19. try { test(); }
20. catch (RuntimeException ex) { System.out.print("runtime "); }
```

```
21. System.out.print("end ");
22. }
```

What is the result?

- A. test end
- B. Compilation fails.
- C. test runtime end
- D. test exception end
- E. A Throwable is thrown by main at runtime.

Answer: D

Q: 161 Given:

```
11. public class Test {
12. public static void main(String [] args) {
13. int x = 5;
14. boolean b1 = true;
15. boolean b2 = false;
16.
17. if ((x == 4) && !b2)
18. System.out.print("1 ");
19. System.out.print("2 ");
20. if ((b2 = true) && b1)
21. System.out.print("3 ");
22. }
23. }
```

What is the result?

- A. 2
- B. 3
- C. 1 2
- D. 2 3
- E. 1 2 3
- F. Compilation fails.
- G. An exception is thrown at runtime.

Answer: D

Q: 162 Click the Exhibit button.

```
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. }
```

Given: ClassA a = new ClassA();

a.methodA();

What is the result?

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

Answer: D

Q: 163 Click the Exhibit button.

```

1. public class A {
2. public void method1() {
3. try {
4. B b = new B();
5. b.method2();
6. // more code here
7. } catch (TestException te) {
8. throw new RuntimeException(te);
9. }
10. }
11. }
12. public class B {
13. public void method2() throws TestException {
14. // more code here
15. }
16. }
17. public class TestException extends Exception {
18. }

```

Given:

```

31. public void method() {
32. A a = new A();
33. a.method1();
34. }

```

Which statement is true if a TestException is thrown on line 3 of class B?

- A. Line 33 must be called within a try block.
- B. The exception thrown by method1 in class A is not required to be caught.
- C. The method declared on line 31 must be declared to throw a RuntimeException.
- D. On line 5 of class A, the call to method2 of class B does not need to be placed in a try/catch block.

Answer: B

Q: 166 Which four statements are true? (Choose four.)

- A. Has-a relationships should never be encapsulated.
- B. Has-a relationships should be implemented using inheritance.
- C. Has-a relationships can be implemented using instance variables.
- D. Is-a relationships can be implemented using the extends keyword.
- E. Is-a relationships can be implemented using the implements keyword.
- F. The relationship between Movie and Actress is an example of an is-a relationship.
- G. An array or a collection can be used to implement a one-to-many has-a relationship.

Answer: C, D, E, G

Q: 168 Given:

```

10. abstract class A {
11. abstract void a1();
12. void a2() { }
13. }
14. class B extends A {
15. void a1() { }
16. void a2() { }

```



```
17. }
18. class C extends B { void c1() { } }
```

and:

```
A x = new B(); C y = new C(); A z = new C();
```

What are four valid examples of polymorphic method calls? (Choose four.)

- A. x.a2();
- B. z.a2();
- C. z.c1();
- D. z.a1();
- E. y.c1();
- F. x.a1();

Answer: A, B, D, F

Q: 169 Given:

```
10. interface Jumper { public void jump(); }
...
```

```
...
```

```
20. class Animal {}
...
```

```
...
```

```
30. class Dog extends Animal {
31. Tail tail;
32. }
...
```

```
...
```

```
...
```

```
...
```

```
40. class Beagle extends Dog implements Jumper{
41. public void jump() {} 42. }
...
```

```
...
```

```
...
```

```
50. class Cat implements Jumper{
51. public void jump() {}
52. }
...
```

```
...
```

```
...
```

Which three are true? (Choose three.)

- A. Cat is-a Animal
- B. Cat is-a Jumper
- C. Dog is-a Animal
- D. Dog is-a Jumper
- E. Cat has-a Animal
- F. Beagle has-a Tail
- G. Beagle has-a Jumper

Answer: B, C, F

Q: 170 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.)

- A. public int blipvert(int x) { return 0; }
- B. private int blipvert(int x) { return 0; }
- C. private int blipvert(long x) { return 0; }
- D. protected long blipvert(int x) { return 0; }
- E. protected int blipvert(long x) { return 0; }
- F. protected long blipvert(long x) { return 0; }
- G. protected long blipvert(int x, int y) { return 0; }

Answer: A, C, E, F, G

Q: 172 Given:

```
10. public class SuperCalc {
11. protected static int multiply(int a, int b) { return a * b; }
...
```

```
...
```



```

12. }
and:
20. public class SubCalc extends SuperCalc{
21. public static int multiply(int a, int b) {
22. int c = super.multiply(a, b);
23. return c;
24. }
25. }
and:
30. SubCalc sc = new SubCalc ();
31. System.out.println(sc.multiply(3,4));
32. System.out.println(SubCalc.multiply(2,2));
What is the result?

```

- A. 12 4
- B. The code runs with no output.
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error in line 21.
- E. Compilation fails because of an error in line 22.
- F. Compilation fails because of an error in line 31.

Answer: E

Q: 173 Given:

```

1. class Pizza {
2. java.util.ArrayList toppings;
3. public final void addTopping(String topping) {
4. toppings.add(topping);
5. }
6. }
7. public class PepperoniPizza extends Pizza {
8. public void addTopping(String topping) {
9. System.out.println("Cannot add Toppings");
10. }
11. public static void main(String[] args) {
12. Pizza pizza = new PepperoniPizza();
13. pizza.addTopping("Mushrooms");
14. }
15. }

```

Given:

```

10. public class Pizza {
11. ArrayList toppings;
12.
13. public final void addTopping(String topping) {
14. toppings.add(topping);
15. }
16.
17. public void removeTopping(String topping) {
18. toppings.remove(topping);
19. }
20. }

```

And:

```
30. class PepperoniPizza extends Pizza {
31. public void addTopping(String topping) {
32. System.out.println("Cannot add Toppings");
33. }
34.
35. public void removeTopping(String topping) {
36. System.out.println("Cannot remove Pepperoni");
37. }
38. }
```

And:

```
50. Pizza pizza = new PepperoniPizza();
51. pizza.addTopping("Mushrooms");
52. pizza.removeTopping("Pepperoni");
```

What is the result?

- A. Compilation fails.
- B. Cannot add Toppings
- C. The code runs with no output.
- D. A NullPointerException is thrown in Line 4.

Answer: A

**Q: 175** 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) { };

Answer: C

**Q: 176 Given:**

```
1. public class MyLogger {
2. private StringBuilder logger = new StringBuuilder();
3. public void log(String message, String user) {
4. logger.append(message);
5. logger.append(user);
6. }
7. }
```

The programmer must guarantee that a single MyLogger object works properly for a multi-threaded system. How must this code be changed to be thread-safe?

- A. synchronize the log method
- B. replace StringBuilder with StringBuffer
- C. replace StringBuilder with just a String object and use the string concatenation (+) within the log method
- D. No change is necessary, the current MyLogger code is already thread-safe.

Answer: A

**Q: 181** Assuming that the serializeBanana2() and the deserializeBanana2() methods will correctly use Java serialization and given:

```
13. import java.io.*;
14. class Food {Food() { System.out.print("1"); } }
```

```

15. class Fruit extends Food implements Serializable {
16. Fruit() { System.out.print("2"); } }
17. public class Banana2 extends Fruit { int size = 42;
18. public static void main(String [] args) {
19. Banana2 b = new Banana2();
20. b.serializeBanana2(b); // assume correct serialization
21. b = b.deserializeBanana2(b); // assume correct
22. System.out.println(" restored " + b.size + " "); }
23. // more Banana2 methods
24. }

```

What is the result?

- A. Compilation fails.
- B. 1 restored 42
- C. 12 restored 42
- D. 121 restored 42
- E. 1212 restored 42
- F. An exception is thrown at runtime.

Answer: D

**Q: 183 Click the Task button.**

Place the code fragments into position to produce the output:

true true false

### Code

```

Scanner scanner = new Scanner("One,5,true,3,true,6,7,false");
scanner.useDelimiter(",");

while () {
 if () {
 System.out.print(+ " ");
 } else ;
}

```

### Code Fragments

Done

**Q: 184 Given:**

```

11. double input = 314159.26;
12. NumberFormat nf = NumberFormat.getInstance(Locale.ITALIAN);
13. String b;
14. //insert code here

```

Which code, inserted at line 14, sets the value of b to 314.159,26?

- A. b = nf.parse( input );
- B. b = nf.format( input );
- C. b = nf.equals( input );
- D. b = nf.parseObject( input );

Answer: B

**Q: 186 Given:**

```

1. public class Score implements Comparable<Score> {

```

```

2. private int wins, losses;
3. public Score(int w, int l) { wins = w; losses = l; }
4. public int getWins() { return wins; }
5. public int getLosses() { return losses; }
6. public String toString() {
7. return "<" + wins + "," + losses + ">";
8. }
9. // insert code here
10. }

```

Which method will complete this class?

- A. public int compareTo(Object o){/\*more code here\*/}
- B. public int compareTo(Score other){/\*more code here\*/}
- C. public int compare(Score s1,Score s2){/\*more code here\*/}
- D. public int compare(Object o1,Object o2){/\*more code here\*/}

Answer: B

Q: 187 Click the Task button.

Place the code in the appropriate places such that this program will always output [1, 2].

```

import java.util.*;

public class MyInt implements Comparable {
 public static void main(String[] args) {
 ArrayList<MyInt> list = new ArrayList<MyInt>();
 list.add(new MyInt(2));
 list.add(new MyInt(1));
 Collections.sort(list);
 System.out.println(list);
 }
 private int i;
 public MyInt(int i) { this.i = i; }
 public String toString() { return Integer.toString(i); }

 public int compareTo(Object o) {
 MyInt i2 = (MyInt)o;
 return i - i2.i;
 }
}

```

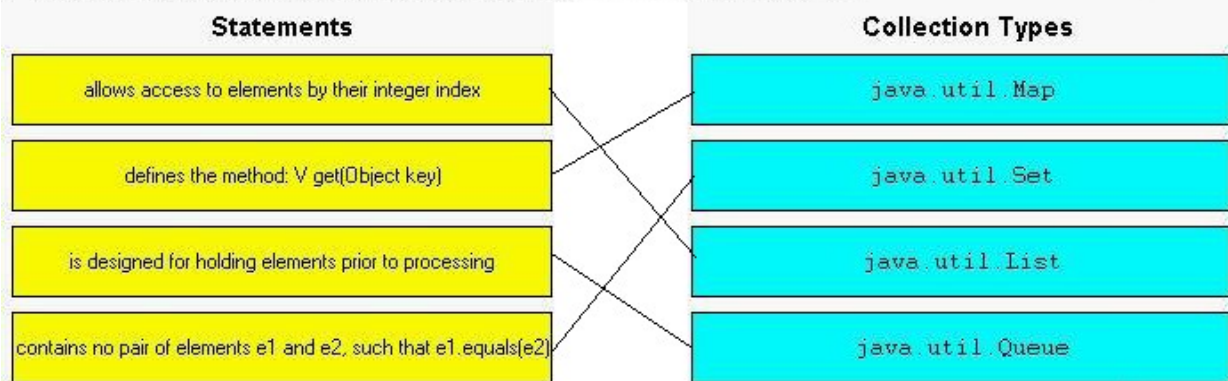
Code

|                            |                              |          |        |            |
|----------------------------|------------------------------|----------|--------|------------|
| implements                 | extends                      | Sortable | Object | Comparable |
| protected                  | public                       | i - i2.i | i      | i2.i - i   |
| compare(MyInt o, MyInt i2) | compare(Object o, Object i2) |          |        |            |
| sort(Object o)             | sort(MyInt o)                |          |        |            |
| compareTo(MyInt o)         | compareTo(Object o)          |          |        |            |

Done

Q: 188 Click the Task button.

Place each Collection Type on the statement to which it applies.



Q: 189 Given:

```

11. public class Person {
12. private String name;
13. public Person(String name) {
14. this.name = name;
15. }
16. public boolean equals(Object o) {
17. if (! o instanceof Person) return false;
18. Person p = (Person) o;
19. return p.name.equals(this.name);
20. }
21. }

```

Which statement is true?

- A. Compilation fails because the hashCode method is not overridden.
- B. A HashSet could contain multiple Person objects with the same name.
- C. All Person objects will have the same hash code because the hashCode method is not overridden.
- D. If a HashSet contains more than one Person object with name="Fred", then removing another Person, also with name="Fred", will remove them all.

Answer: B

Q: 190 Given:

```

13. public static void search(List<String> list) {
14. list.clear();
15. list.add("b");
16. list.add("a");
17. list.add("c");
18. System.out.println(Collections.binarySearch(list, "a"));
19. }

```

What is the result of calling search with a valid List implementation?

- A. 0
- B. 1
- C. 2
- D. a
- E. b
- F. c
- G. The result is undefined.

Answer: G

Q: 191 Click the Task button.

Place the code into the GenericB class definition to make the class compile successfully.

```
import java.util.*;
```

```
public class GenericB< T extends Pet. > {
 public T foo;
 public void setFoo(T foo) {
 this.foo = foo;
 }
 public T getFoo() {
 return foo;
 }
 public static void main (String[] args) {
 GenericB<Cat> bar = new GenericB<Cat>();
 bar.setFoo(new Cat());
 Cat c = bar.getFoo();
 }
}
```

```
interface Pet { }
class Cat implements Pet{ }
```

### Code

? extends Pet

T extends Pet

? implements Pet

T implements Pet

Pet extends T

?

T

<?>

Pet

Done

Q: 192 Which two statements are true about the hashCode method? (Choose two.)

- A. The hashCode method for a given class can be used to test for object equality and object inequality for that class.
- B. The hashCode method is used by the java.util.SortedSet collection class to order the elements within that set.
- C. The hashCode method for a given class can be used to test for object inequality, but NOT object equality, for that class.
- D. The only important characteristic of the values returned by a hashCode method is that the distribution of values must follow a Gaussian distribution.
- E. The hashCode method is used by the java.util.HashSet collection class to group the elements within that set into hash buckets for swift retrieval.

Answer: C, E

Q: 196 Given:

1. package com.company.application;
- 2.
3. public class MainClass {
4. public static void main(String[] args) {}
5. }

And MainClass exists in the /apps/com/company/application directory. Assume the CLASSPATH environment variable is set to "." (current directory).

Which two java commands entered at the command line will run MainClass?(Choosetwo)

- A. java MainClass if run from the /apps directory
- B. java com.company.application.MainClass if run from the /apps directory
- C. java -classpath /apps com.company.application.MainClass if run from any directory
- D. java -classpath . MainClass if run from the /apps/com/company/application directory
- E. java -classpath /apps/com/company/application:. MainClass if run from the /apps directory
- F. java com.company.application.MainClass if run from the /apps/com/company/application directory

Answer: B, C

**Q: 197 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

**Q: 198** 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

Q: 200 Click the Task button.



Place code fragments into position so the output is: The quantity is 420

```
public int update(int quantity, int adjust) {
 quantity = quantity + adjust;
 return quantity;
}

public void callUpdate() {
 int quant = 100;
 quant = update(quant, 320);
 System.out.println("The quantity is " + quant);
}
```

### Code Fragments

|             |                               |                                                   |
|-------------|-------------------------------|---------------------------------------------------|
| public int  | quantity = quantity + adjust; | update(quant, 320);                               |
| public void | quant = update(quant, 320);   | quantity = quantity + adjust;<br>return quantity; |

Q: 201 Given the command line java Pass2 and:

```
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. }
```

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

Q: 203 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: 204 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: 206** A UNIX user named Bob wants to replace his chess program with a new one, but he is not sure where the old

one is installed. Bob is currently able to run a Java chess program starting from his home directory /home/bob using the command:

```
java -classpath /test:/home/bob/downloads/*.jar games.Chess
```

Bob's CLASSPATH is set (at login time) to:

```
/usr/lib:/home/bob/classes:/opt/java/lib:/opt/java/lib/*.jar
```

What is a possible location for the Chess.class file?

- A. /test/Chess.class
- B. /home/bob/Chess.class
- C. /test/games/Chess.class
- D. /usr/lib/games/Chess.class
- E. /home/bob/games/Chess.class
- F. inside jarfile /opt/java/lib/Games.jar (with a correct manifest)
- G. inside jarfile /home/bob/downloads/Games.jar (with a correct manifest)

Answer: C

Q: 207 Click the Exhibit button.

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());

```
1. public class A {
2.
3. private int counter = 0;
4.
5. public static int getInstanceCount() {
6. return counter;
7. }
8.
9. public A() {
10. counter++;
11. }
12.
13. }
```

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: 208 Click the Exhibit button.

```
1. public class GoTest {
2. public static void main(String[] args)
3. {
4. Sente a = new Sente(); a.go();
5. Goban b = new Goban(); b.go();
6. Stone c = new Stone(); c.go();
7. }
8.
9. class Sente implements Go {
10. public void go() {
11. System.out.println("go in Sente.");
12. }
13. }
14. class Goban extends Sente {
15. public void go() {
16. System.out.println("go in Goban");
17. }
18. }
19. interface Go { public void go(); }
```

What is the result?

- A. go in Goban  
go in Sente
- B. go in Sente  
go in Goban
- C. go in Goban  
go in Sente
- D. Compilation fails because of an error in line 17.

Answer: B

Q: 209 Click the Exhibit button.

Given:

```

25. A a = new A();
26. System.out.println(a.doit(4, 5));

1. public class A {
2. public String doit(int x, int y) {
3. return "a";
4. }
5.
6. public String doit(int... vals) {
7. return "b";
8. }
9. }

```

What is the result?

- A. Line 26 prints "a" to System.out.
- B. Line 26 prints "b" to System.out.
- C. An exception is thrown at line 26 at runtime.
- D. Compilation of class A will fail due to an error in line 6.

Answer: A

Q: 210 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?

- A. TestA
- B. TestB
- C. Compilation fails.
- D. An exception is thrown at runtime.

Answer: B

Q: 211 Given:

```

35. String #name = "Jane Doe";
36. int $age = 24;
37. Double _height = 123.5;
38. double ~temp = 37.5;

```

Which two statements are true? (Choose two.)

- A. Line 35 will not compile.
- B. Line 36 will not compile.
- C. Line 37 will not compile.
- D. Line 38 will not compile.

Answer: A, D

Q: 213 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?

- A. test
- B. null
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error in line 1.
- E. Compilation fails because of an error in line 4.
- F. Compilation fails because of an error in line 5.

Answer: A

Q: 215 Given

```
11. public interface Status {
12. /* insert code here */ int MY_VALUE = 10;
13. }
```

Which three are valid on line 12? (Choose three.)

- A. final
- B. static
- C. native
- D. public
- E. private
- F. abstract
- G. protected

Answer: A, B, D

Q: 216 Given:

```
55. int [] x = {1, 2, 3, 4, 5};
56. int y[] = x;
57. System.out.println(y[2]);
```

Which statement is true?

- A. Line 57 will print the value 2.
- B. Line 57 will print the value 3.
- C. Compilation will fail because of an error in line 55.
- D. Compilation will fail because of an error in line 56.

Answer: B

Q: 218 A programmer needs to create a logging method that can accept an arbitrary number of arguments. For example, it may be called in these ways:

```
logIt("log message1");
logIt("log message2", "log message3");
logIt("log message4", "log message5", "log message6");
```

Which declaration satisfies this requirement?

- A. public void logIt(String \* msgs)
- B. public void logIt(String [] msgs)
- C. public void logIt(String... msgs)
- D. public void logIt(String msg1, String msg2, String msg3)

Answer: C

Q: 220 Click the Task button.

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 void produce() {
 isReady = true;
 ;
 }

 public void consume() {
 while (! isReady) {
 try {
 ;
 } catch (Exception ex) { }
 }
 isReady = ;
 }
}
```

#### Code Elements

|                                           |                                             |                                          |                                          |
|-------------------------------------------|---------------------------------------------|------------------------------------------|------------------------------------------|
| <input type="text" value="synchronized"/> | <input type="text" value="true"/>           | <input type="text" value="false"/>       | <input type="text" value="wait()"/>      |
| <input type="text" value="volatile"/>     | <input type="text" value="synchronized()"/> | <input type="text" value="notifyAll()"/> | <input type="text" value="synchronize"/> |

Q: 221 Given:

1. public class Threads4 {
2. public static void main (String[] args) {
3. new Threads4().go();
4. }
5. public void go() {
6. Runnable r = new Runnable() {
7. public void run() {
8. System.out.print("foo");
9. }
10. };
11. Thread t = new Thread(r);
12. t.start();
13. t.start();
14. }
15. }

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes normally and prints "foo".
- D. The code executes normally, but nothing is printed.

Answer: B

Q: 223 Click the Task button.

Place the code elements into the class so that the code compiles and prints "Run. Run. doIt." in exactly that order. Note that there may be more than one correct solution.

```
public class TestTwo extends Thread {
 public static void main (String[] a) throws Exception {
 TestTwo t = new TestTwo();
 t.start();
 t.join();
 t.run();
 t.doIt();
 }
 public void run() {
 System.out.print("Run. ");
 }
 public void doIt() {
 System.out.print("doIt. ");
 }
}
```

**Code Elements**

**Q: 224 Given:**

```
1. public class Threads5 {
2. public static void main (String[] args) {
3. new Thread(new Runnable() {
4. public void run() {
5. System.out.print("bar");
6. }).start();
7. }
8. }
```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes normally and prints "bar".
- D. The code executes normally, but nothing prints.

Answer: C

**Q: 231 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?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The attribute id in the Item object remains unchanged.
- D. The attribute id in the Item object is modified to the new value.
- E. A new Item object is created with the preferred value in the id attribute.

Answer: A

**Q: 235 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: 240 A programmer has an algorithm that requires a java.util.List that provides an efficient implementation of add(0, object), but does NOT need to support quick random access. What supports these requirements?

- A. java.util.Queue
- B. java.util.ArrayList
- C. java.util.LinearList
- D. java.util.LinkedList

Answer: D

**Q: 243 Click the Task button.**

Given the class definitions:

```
class Animal { }
class Dog extends Animal { }
```

and the code:

```
public void go() {
 ArrayList<Dog> aList = new ArrayList<Dog>();
 takeList(aList);
}
// insert definition of the takeList() method here
```

Place the correct Compilation Result on each takeList() method definition to indicate whether or not the go() method would compile given that definition.

**takeList() Method Definition**

|                                                            |                       |
|------------------------------------------------------------|-----------------------|
| public void takeList(ArrayList list) { }                   | Compilation succeeds. |
| public void takeList(ArrayList<Animal> list) { }           | Compilation fails.    |
| public void takeList(ArrayList<? extends Animal> list) { } | Compilation succeeds. |
| public void takeList(ArrayList<?> list) { }                | Compilation succeeds. |
| public void takeList(ArrayList<Object> list) { }           | Compilation fails.    |

**Compilation Result**

|                       |
|-----------------------|
| Compilation succeeds. |
| Compilation fails.    |

**Q: 245 Given:**

```
1. public class Drink implements Comparable {
2. public String name;
3. public int compareTo(Object o) {
4. return 0;
5. }
6. }
```



and:

```
20. Drink one = new Drink();
21. Drink two = new Drink();
22. one.name= "Coffee";
23. two.name= "Tea";
23. TreeSet set = new TreeSet();
24. set.add(one);
25. set.add(two);
```

A programmer iterates over the TreeSet and prints the name of each Drink object.

What is the result?

- A. Tea
- B. Coffee
- C. CoffeeTea
- D. Compilation fails.
- E. The code runs with no output.
- F. An exception is thrown at runtime.

Answer: B

Q: 246 Click the Task button.

Given:

```
1. import java.util.*;
2. public class TestGenericConversion {
3. public static void main(String[] args) {
4. List list = new LinkedList();
5. list.add("one");
6. list.add("two");
7. System.out.print(((String)list.get(0)).length());
8. }
9. }
```

Refactor this class to use generics without changing the code's behavior.

```
1. import java.util.*;
2. public class TestGenericConversion {
3. public static void main(String[] args) {
4.
5. List<String> list = new LinkedList<String>();
6. list.add("one");
7. list.add("two");
8. System.out.print(list.get(0).length());
9. }
10. }
```

Code

|                                               |                                                           |
|-----------------------------------------------|-----------------------------------------------------------|
| List list = new LinkedList();                 | System.out.print( list.get(0).length() );                 |
| List<String> list = new LinkedList<String>(); | System.out.print( list.get<String>(0).length() );         |
| List<String> list = new LinkedList();         | System.out.print( <String>list.get(0).length() );         |
| List list = new LinkedList<String>();         | System.out.print( ((List<String>)list.get(0)).length() ); |

Q: 249 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: 250 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 javaConverter 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: 251 Click the Task button.

Place the Fragments into the program, so that the program will get lines from a text file, display them, and then close all the resources.

### Program

```
import java.io.*

public class ReadFile {
 public static void main(String [] args) {
 try {
 File x1 = new File("MyText.txt");
 FileReader x2 = new FileReader(x1);
 BufferedReader x4 = new BufferedReader(x2);
 String x3 = null;
 while ((x3 = x4. Place here ()) != null) {
 System.out.println(x3);
 } x4. close ();
 } catch (Exception ex) {
 ex.printStackTrace();
 }
 }
}
```

### Code Fragments

BufferedReader  
 StreamReader  
 FileReader  
 readLine  
 readLn  
 read  
 closeFile  
 close  
 x1 x2  
 x3 x4

Done

Q: 255 Given:

```
12. String csv = "Sue,5,true,3";
13. Scanner scanner = new Scanner(csv);
14. scanner.useDelimiter(",");
15. int age = scanner.nextInt();
```

What is the result?

- A. Compilation fails.
- B. After line 15, the value of age is 5.
- C. After line 15, the value of age is 3.
- D. An exception is thrown at runtime.

Answer: D

Q: 257 Given:

```
11. String test = "a1b2c3";
12. String[] tokens = test.split("\\d");
13. for(String s: tokens) System.out.print(s + " ");
```

What is the result?

- A. a b c
- B. 1 2 3
- C. a1b2c3
- D. a1 b2 c3
- E. Compilation fails.
- F. The code runs with no output.
- G. An exception is thrown at runtime.

Answer: A

Q: 258 Given:

```
10. class One {
11. public One() { System.out.print(1); }
12. }
13. class Two extends One {
14. public Two() { System.out.print(2); }
15. }
16. class Three extends Two {
17. public Three() { System.out.print(3); }
18. }
19. public class Numbers{
20. public static void main(String[] argv) { new Three(); }
21. }
```

What is the result when this code is executed?

- A. 1
- B. 3
- C. 123
- D. 321
- E. The code runs with no output.

Answer: C

Q: 260 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: 262 Given:

```
10. class One {
11. void foo() { }
12. }
13. class Two extends One {
14. //insert method here
15. }
```

Which three methods, inserted individually at line 14, will correctly complete class Two? (Choose three.)

- A. int foo() { /\* more code here \*/ }
- B. void foo() { /\* more code here \*/ }
- C. public void foo() { /\* more code here \*/ }
- D. private void foo() { /\* more code here \*/ }
- E. protected void foo() { /\* more code here \*/ }

Answer: B, C, E

Q: 263 Given:

```
10. interface Data { public void load(); }
11. abstract class Info { public abstract void load(); }
```

Which class correctly uses the Data interface and Info class?

- A. public class Employee extends Info implements Data {  
public void load() { /\*do something\*/ } }
- B. public class Employee implements Info extends Data {  
public void load() { /\*do something\*/ } }
- C. public class Employee extends Info implements Data  
public void load(){ /\*do something\*/ }  
public void Info.load(){ /\*do something\*/ } }
- D. public class Employee implements Info extends Data {  
public void Data.load(){ /\*do something\*/ }  
public void load(){ /\*do something\*/ } }
- E. public class Employee implements Info extends Data {  
public void load(){ /\*do something\*/ }  
public void Info.load(){ /\*do something\*/ } }
- F. public class Employee extends Info implements Data{  
public void Data.load() { /\*do something\*/ }  
public void Info.load() { /\*do something\*/ } }

Answer: A

Q: 264 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: 265 Given:

```
11. public abstract class Shape {
```

```

12. int x;
13. int y;
14. public abstract void draw();
15. public void setAnchor(int x, int y) {
16. this.x = x;
17. this.y = y;
18. }
19. }

```

and a class Circle that extends and fully implements the Shape class.  
Which is correct?

- A. Shape s = new Shape();  
s.setAnchor(10,10);  
s.draw();
- B. Circle c = new Shape();  
c.setAnchor(10,10);  
c.draw();
- C. Shape s = new Circle();  
s.setAnchor(10,10);  
s.draw();
- D. Shape s = new Circle();  
s->setAnchor(10,10);  
s->draw();
- E. Circle c = new Circle();  
c.Shape.setAnchor(10,10);  
c.Shape.draw();

Answer: C

Q: 268 Click the Exhibit button.

```

11. class Person {
12. String name = "No name";
13. public Person(String nm) { name = nm; }
14. }
15.
16. class Employee extends Person {
17. String empID = "0000";
18. public Employee(String id) { empID =
19. id; }
20. }
21. public class EmployeeTest {
22. public static void main(String[] args)
23. {
24. Employee e = new Employee("4321");
25. System.out.println(e.empID);
26. }

```

What is the result?

- A. 4321
- B. 0000
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error in line 18.

Answer: D

Q: 269 Which two code fragments correctly create and initialize a static array of int elements? (Choose two.)

- A. static final int[] a = { 100,200 };
- B. static final int[] a;  
static { a=new int[2]; a[0]=100; a[1]=200; }

- C. static final int[] a = new int[2]{ 100,200 };
- D. static final int[] a;  
static void init() { a = new int[3]; a[0]=100; a[1]=200; }

Answer: A,B

Q: 272 Click the Exhibit button.

SomeException:

```
1. public class SomeException {
2. }
```

Class A:

```
1. public class A {
2. public void doSomething() { }
3. }
```

Class B:

```
1. public class B extends A {
2. public void doSomething() throws
SomeException { }
3. }
```

Which statement is true about the two classes?

- A. Compilation of both classes will fail.
- B. Compilation of both classes will succeed.
- C. Compilation of class A will fail. Compilation of class B will succeed.
- D. Compilation of class B will fail. Compilation of class A will succeed.

Answer: D

Q: 273 Given:

```
11. public static void main(String[] args) {
12. for (int i = 0; i <= 10; i++) {
13. if (i > 6) break;
14. }
15. System.out.println(i);
16. }
```

What is the result?

- A. 6
- B. 7
- C. 10
- D. 11
- E. Compilation fails.
- F. An exception is thrown at runtime.

Answer: E

Q: 277 Given:

```
11. public static void main(String[] args) {
12. Integer i = new Integer(1) + new Integer(2);
13. switch(i) {
14. case 3: System.out.println("three"); break;
15. default: System.out.println("other"); break;
16. }
17. }
```

What is the result?

- A. three

- B. other
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error on line 12.
- E. Compilation fails because of an error on line 13.
- F. Compilation fails because of an error on line 15.

Answer: A

Q: 278 Given:

```
10. public class ClassA {
11. public void count(int i) {
12. count(++i);
13. }
14. }
```

And:

```
20. ClassA a = new ClassA();
21. a.count(3);
```

Which exception or error should be thrown by the virtual machine?

- A. StackOverflowError
- B. NullPointerException
- C. NumberFormatException
- D. IllegalArgumentException
- E. ExceptionInInitializerError

Answer: A

Q: 283 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: 284 Given:

```
10. interface A { void x(); }
11. class B implements A { public void x() {} public void y() {} }
12. class C extends B { public void x() {} }
```

And:

```
20. java.util.List<A> list = new java.util.ArrayList<A>();
21. list.add(new B());
22. list.add(new C());
23. for (A a : list) {
24. a.x();
25. a.y();
26. }
```

What is the result?

- A. The code runs with no output.
- B. An exception is thrown at runtime.
- C. Compilation fails because of an error in line 20.
- D. Compilation fails because of an error in line 21.
- E. Compilation fails because of an error in line 23.
- F. Compilation fails because of an error in line 25.

Answer: F

Q: 285 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?

- A. Compilation will succeed if A extends B.
- B. Compilation will succeed if B extends A.
- C. Compilation will always fail because of an error in line 7.
- D. Compilation will always fail because of an error in line 8.

Answer: B

Q: 287 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;
public class Alpha {
 private String alpha;
 public Alpha() { this("A"); }
 protected Alpha(String a) { alpha = a; }
}
```

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

#### Modifiers

**private**

**protected**

**public**

Done

Q: 291 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: 293 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: 297 Given:**

```
35. int x = 10;
36. do {
37. x--;
38. } while (x < 10);
How many times will line 37 be executed?
```

- A. ten times
- B. zero times
- C. one to nine times
- D. more than ten times

Answer: D

**Q: 300 Given:**

```
11. static class A {
12. void process() throws Exception { throw new Exception(); }
13. }
14. static class B extends A {
15. void process() { System.out.println("B"); }
16. }
17. public static void main(String[] args) {
18. new B().process();
19. }
```

What is the result?

- A. B
- B. The code runs with no output.
- 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 18.

Answer: A

**Q: 301 Given:**

```
11. static void test() {
12. try {
13. String x = null;
14. System.out.print(x.toString() + " ");
15. }
16. finally { System.out.print("finally "); }
17. }
18. public static void main(String[] args) {
19. try { test(); }
20. 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

Q: 306 Given:

```
11. class ClassA {}
12. class ClassB extends ClassA {}
13. class ClassC extends ClassA {}
```

and:

```
21. ClassA p0 = new ClassA();
22. ClassB p1 = new ClassB();
23. ClassC p2 = new ClassC();
24. ClassA p3 = new ClassB();
25. ClassA p4 = new ClassC();
```

Which three are valid? (Choose three.)

- A. p0 = p1;
- B. p1 = p2;
- C. p2 = p4;
- D. p2 = (ClassC)p1;
- E. p1 = (ClassB)p3;
- F. p2 = (ClassC)p4;

Answer: A, E, F

Q: 307 Given:

```
11. abstract class Vehicle { public int speed() { return 0; }
12. class Car extends Vehicle { public int speed() { return 60; }
13. class RaceCar extends Car { public int speed() { return 150; }
...
```

```
21. RaceCar racer = new RaceCar();
22. Car car = new RaceCar();
23. Vehicle vehicle = new RaceCar();
24. System.out.println(racer.speed() + ", " + car.speed()
25. + ", " + vehicle.speed());
```

What is the result?

- A. 0, 0, 0
- B. 150, 60, 0
- C. Compilation fails.
- D. 150, 150, 150
- E. An exception is thrown at runtime.

Answer: D

Q: 308 Click the Task button.

Given: 

```
public class Doubler {
 public static int doubleMe(Holder h) {
 return h.getAmount() * 2;
 }
}
```

and: 

```
public class Holder {
 int amount = 10;
 public void doubleAmount(){ amount = Doubler.doubleMe(this);}
 public int getAmount(){ return amount;}
 //more code here
}
```

Place the code fragments in position to reduce the coupling between Doubler and Holder.

```
public class Doubler {
 public static int doubleMe(int h) {
 return h * 2;
 }
}
```

```
public class Holder {
 int amount = 10;
 public void doubleAmount(){ amount = Doubler.doubleMe(amount);}
 public int getAmount(){ return amount;}
 //more code here
}
```

#### Code Fragments

|               |        |      |         |
|---------------|--------|------|---------|
| void          | Holder | int  | Doubler |
| h.getAmount() | h      | this | amount  |

Done

Q: 309 Given code in separate source files:

```
10. public class Foo {
11. public int a;
12. public Foo() { a = 3; }
13. public void addFive() { a += 5;}
14. } and: 20. public class Bar extends Foo {
21. public int a;
22. public Bar() { a = 8; }
23. public void addFive() { this.a += 5; }
24. } invoked with:
30. Foo foo = new Bar();
31. foo.addFive();
32. System.out.println("Value: " + foo.a);
```

What is the result?

- A. Value: 3
- B. Value: 8
- C. Value: 13
- D. Compilation fails.
- E. The code runs with no output.
- F. An exception is thrown at runtime.

Answer: A

**Q: 310** Which Man class properly represents the relationship "Man has a best friend who is a Dog"?

- A. class Man extends Dog { }
- B. class Man implements Dog { }
- C. class Man { private BestFriend dog; }
- D. class Man { private Dog bestFriend; }

E. class Man { private Dog<bestFriend>; }  
F. class Man { private BestFriend<dog>; }

Answer: D

Q: 317 Given:

```
10. public class MyClass {
11.
12. public Integer startingI;
13. public void methodA() {
14. Integer i = new Integer(25);
15. startingI = i;
16. methodB(i);
17. }
18. private void methodB(Integer i2) {
19. i2 = i2.intValue();
20.
21. }
22. }
```

If methodA is invoked, which two are true at line 20? (Choose two.)

- A. i2 == startingI returns true.
- B. i2 == startingI returns false.
- C. i2.equals(startingI) returns true.
- D. i2.equals(startingI) returns false.

Answer: B, C

Q: 320 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?

- A. text
- B. An exception is thrown at runtime.
- C. Compilation fails because of an error in line 2 of XMLMessage.
- D. Compilation fails because of an error in line 3 of XMLMessage.

Answer: D

Q: 342 Given:

```
10. class Line {
11. public class Point { public int x,y;}
12. public Point getPoint() { return new Point(); }
13. }
14. class Triangle {
15. public Triangle() {
16. // insert code here
17. }
18. }
```

Which code, inserted at line 16, correctly retrieves a local instance of a Point object?

- A. Point p = Line.getPoint();
- B. Line.Point p = Line.getPoint();
- C. Point p = (new Line()).getPoint();
- D. Line.Point p = (new Line()).getPoint();

Answer: D

**Q: 349 Given:**

```
10. interface Foo { int bar(); }
11. public class Sprite {
12. public int fubar(Foo foo) { return foo.bar(); }
13. public void testFoo() {
14. fubar(
15. // insert code here
16.);
17. }
18. }
```

Which code, inserted at line 15, allows the class Sprite to compile?

- A. Foo { public int bar() { return 1; } }
- B. new Foo { public int bar() { return 1; } }
- C. new Foo() { public int bar() { return 1; } }
- D. new class Foo { public int bar() { return 1; } }

Answer: C

**Q: 350 Given:**

```
1. public class A {
2. public void doit() {
3. }
4. public String doit() {
5. return "a";
6. }
7. public double doit(int x) {
8. return 1.0;
9. }
10. }
```

What is the result?

- A. An exception is thrown at runtime.
- B. Compilation fails because of an error in line 7.
- C. Compilation fails because of an error in line 4.
- D. Compilation succeeds and no runtime errors with class A occur.

Answer: C

**Q: 351 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: 352 Given:

```
10. abstract public class Employee {
11. protected abstract double getSalesAmount();
12. public double getCommision() {
13. return getSalesAmount() * 0.15;
14. }
15. }
16. class Sales extends Employee {
17. // insert method here
18. }
```

Which two methods, inserted independently at line 17, correctly complete the Sales class? (Choose two.)

- A. double getSalesAmount() { return 1230.45; }
- B. public double getSalesAmount() { return 1230.45; }
- C. private double getSalesAmount() { return 1230.45; }
- D. protected double getSalesAmount() { return 1230.45; }

Answer: B, D

Q: 355 Given:

```
11. public static void append(List list) { list.add("0042"); }
12. public static void main(String[] args) {
13. List<Integer> intList = new ArrayList<Integer>();
14. append(intList);
15. System.out.println(intList.get(0));
16. }
```

What is the result?

- A. 42
- B. 0042
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error in line 13.
- E. Compilation fails because of an error in line 14.

Answer: B

Q: 357 Given:

```
11. public class Person {
12. private name;
13. public Person(String name) {
14. this.name = name;
15. }
16. public int hashCode() {
17. return 420;
18. }
19. }
```

Which statement is true?

- A. The time to find the value from HashMap with a Person key depends on the size of the map.
- B. Deleting a Person key from a HashMap will delete all map entries for all keys of type Person.
- C. Inserting a second Person object into a HashSet will cause the first Person object to be removed as a duplicate.
- D. The time to determine whether a Person object is contained in a HashSet is constant and does NOT depend on the size of the map.

Answer: A

**Q: 360** A programmer must create a generic class MinMax and the type parameter of MinMax must implement Comparable. Which implementation of MinMax will compile?

- A. 

```
class MinMax<E extends Comparable<E>> {
 E min = null;
 E max = null;
 public MinMax() {}
 public void put(E value) { /* store min or max */ }
}
```
- B. 

```
class MinMax<E implements Comparable<E>> {
 E min = null;
 E max = null;
 public MinMax() {}
 public void put(E value) { /* store min or max */ }
}
```
- C. 

```
class MinMax<E extends Comparable<E>> {
 <E> E min = null;
 <E> E max = null;
 public MinMax() {}
 public <E> void put(E value) { /* store min or max */ }
}
```
- D. 

```
class MinMax<E implements Comparable<E>> {
 <E> E min = null;
 <E> E max = null;
 public MinMax() {}
 public <E> void put(E value) { /* store min or max */ }
}
```

Answer: A

**Q: 361** Given:

```
int[] myArray = new int[] {1, 2, 3, 4, 5};
```

What allows you to create a list from this array?

- A. 

```
List myList = myArray.asList();
```
- B. 

```
List myList = Arrays.asList(myArray);
```
- C. 

```
List myList = new ArrayList(myArray);
```
- D. 

```
List myList = Collections.fromArray(myArray);
```

Answer: B

**Q: 362** Click the Task button.

Place each Collection Type on its function. Note: Not all functions will be used.

| Function                                             | Collection Type                   |
|------------------------------------------------------|-----------------------------------|
| provides array manipulation utilities                | <code>java.util.SortedSet</code>  |
| provides collection manipulation utilities           | <code>java.util.Arrays</code>     |
| defines base methods for all array objects           | <code>java.util.Iterator</code>   |
| defines base methods for all collection objects      | <code>java.util.TreeSet</code>    |
| provides a concrete implementation of an ordered set | <code>java.util.Collection</code> |
| defines base methods for an ordered set              |                                   |
| defines methods for linear access to a collection    |                                   |
| defines methods for random access to a collection    |                                   |

Done