Outputs produced by copying outputs of files.

First Code Trace:

Output:

14
writer
0
writer
0
pen
6
writer
33
pen
15
writer
0
pen

Multiple Choice:

1. e
2. a
3. SKIP
4. SKIP
5. d
6. a
7. d
8. b
9. e
10. a
11. a
12. b
13. b
14. a
15. d
16. b
17. SKIP

Second Code Trace:

Output:

bye
success
bye
success
NPE

Multiple Choice:

18. a
19. b
20. a
21. b
22. e

23. Imagine the type Box extends the type Shape. What must be true for the Box type?
a. Box must be a concrete class
b. If Shape is a class, Box can call Shape's constructor(s) through the super keyword
c. A and B

Answer: b. It isn't necessary for Box to be concrete. Abstract classes and Interfaces can extend as well. However, if Shape is a class, Box needs to be able to get the Shape constructor and it has to do that through super.

24. If you see the code Box.name in a method of a class other than Box, what must be true about name?
a. It must be public
b. It must be static
c. A and B
d. None of the above

Answer: c. It's static because you can access the member from the class name. It's public because you access the member without using a getter.

25. If you see the following code in a method of a class other than Box, what must be true about foo?

```
Box b1 = new Box();
b1.foo();
```

a. It must be public
b. It must be static
c. A and B
d. None of the above

Answer: a. It just has to be public (able to be used outside of Box). It *can* be used if it were static, but it isn't guaranteed.

26. Imagine I have a class Box that implements the Comparable interface. What must be true about Box?
a. It must be an abstract class if it does not have a compareTo method.
b. Writing code for the method with signature public int compareTo(Box b) implements the Comparable interface.
c. Box cannot implements any other interfaces.
d. A and B
e. A and C

Answer: d.

27. Imagine I have a static method foo inside the class Box. What must be true about foo? Assume no methods are called on objects in Box.
a. foo can only call other static methods of Box
b. foo can only use static attributes of Box
c. A and B
d. None of the above.

Answer: d.

28. Imagine I have the compiling code. What must be true?

```
Box b = new Box(6);
```

a. Box cannot have a default constructor.
b. Box must store an integer as an attribute.
c. Box cannot be an abstract class
d. A and C
e. B and C

Answer: c.

29. Imagine a file called Box.java has the following code, what cannot be true about Box?:

```
public void foo();
```

a. Box is an interface
b. Box is an abstract class
c. Box is a concrete class
d. B and C

Answer: d. It is possible to have an abstract method in an abstract class if the signature were "public abstract void foo();"

30. The following class declaration would compile (assume all imports exist and the rest of the class is properly defined between curly braces):

```
public class Box implements Comparable, Iterator
```

a. True b. False

Answer: True. You can implement many but extend once.

31. An unchecked exception must be caught by a try-catch block, in order for Java to compile the code.

a. True b. False

Answer: False. Unchecked exceptions are bugs in the code. They are typically Runtime Exceptions, therefore implying that it will *already have compiled* by the time it has been discovered. By contrast, checked exceptions are things that are things that go wrong outside of the program like if the user is asked to put in a number and they go "Derp. I'm going to type my name...".

32. If Box is a child class of Shape, the assignment Box b1 = new Shape(); would compile.

a. True b. False

Answer: False. It doesn't compile.

33. If Box is a child class of Shape, the assignment Box b1 = (Box) new Shape(); would compile.

a. True b. False

Answer: True. It would compile. However, at run time, it would throw a ClassCastException