

Java Standard Edition 6 Programmer Certified Professional Exam - Mock Exam IV

Section 1: Declarations, Initialization and Scoping

- Develop code that declares classes (including abstract and all forms of nested classes), interfaces, and enums, and includes the appropriate use of package and import statements (including static imports).
- Develop code that declares an interface. Develop code that implements or extends one or more interfaces.
- Develop code that declares an abstract class. Develop code that extends an abstract class.
- Develop code that declares, initializes, and uses primitives, arrays, enums, and objects as static, instance, and local variables. Also, use legal identifiers for variable names.
- Given a code example, determine if a method is correctly overriding or overloading another method, and identify legal return values (including covariant returns), for the method.
- Given a set of classes and superclasses, develop constructors for one or more of the classes. Given a class declaration, determine if a default constructor will be created, and if so, determine the behavior of that constructor. Given a nested or non-nested class listing, write code to instantiate the class.

Section 2: Flow Control

- Develop code that implements an if or switch statement; and identify legal argument types for these statements.
- Develop code that implements all forms of loops and iterators, including the use of for, the enhanced for loop (for-each), do, while, labels, break, and continue; and explain the values taken by loop counter variables during and after loop execution.
- Develop code that makes use of assertions, and distinguish appropriate from inappropriate uses of assertions.
- Develop code that makes use of exceptions and exception handling clauses (try, catch, finally), and declares methods and overriding methods that throw exceptions.
- Recognize the effect of an exception arising at a specified point in a code fragment. Note that the exception may be a runtime exception, a checked exception, or an error.
- Recognize situations that will result in any of the following being thrown: `ArrayIndexOutOfBoundsException`, `ClassCastException`, `IllegalArgumentException`, `IllegalStateException`, `NullPointerException`, `NumberFormatException`, `AssertionError`, `ExceptionInInitializerError`, `StackOverflowError` or `NoClassDefFoundError`. Understand which of these are thrown by the virtual machine and recognize situations in which others should be thrown programmatically.

Section 3: API Contents

- Develop code that uses the primitive wrapper classes (such as `Boolean`, `Character`, `Double`, `Integer`, etc.), and/or autoboxing & unboxing. Discuss the differences between the `String`, `StringBuilder`, and `StringBuffer` classes.
- Given a scenario involving navigating file systems, reading from files, writing to files, or interacting with the user, develop the correct solution using the following classes (sometimes in combination), from `java.io`: `BufferedReader`, `BufferedWriter`, `File`, `FileReader`, `FileWriter`, `PrintWriter`, and `Console`.
- Use standard J2SE APIs in the `java.text` package to correctly format or parse dates, numbers, and currency values for a specific locale; and, given a scenario, determine the appropriate methods to use if you want to use the default locale or a specific locale. Describe the purpose and use of the `java.util.Locale` class.
- Write code that uses standard J2SE APIs in the `java.util` and `java.util.regex` packages to format or parse strings or streams. For strings, write code that uses the `Pattern` and `Matcher` classes and the `String.split` method. Recognize and use regular expression patterns for matching (limited to: `.` (dot), `*` (star), `+` (plus), `?`, `\d`, `\s`, `\w`, `[]`, `()`). The use of `*`, `+`, and `?` will be limited to greedy quantifiers, and the parenthesis operator will only be used as a grouping mechanism, not for capturing content during matching. For streams, write code using the `Formatter` and `Scanner` classes and the `PrintWriter.format/print` methods. Recognize and use formatting parameters (limited to: `%b`, `%c`, `%d`, `%f`, `%s`) in format strings.

Section 4: Concurrency

- Write code to define, instantiate, and start new threads using both `java.lang.Thread` and `java.lang.`

Runnable.

- Recognize the states in which a thread can exist, and identify ways in which a thread can transition from one state to another.
- Given a scenario, write code that makes appropriate use of object locking to protect static or instance variables from concurrent access problems.

Section 5: OO Concepts

- Develop code that implements tight encapsulation, loose coupling, and high cohesion in classes, and describe the benefits.
- Given a scenario, develop code that demonstrates the use of polymorphism. Further, determine when casting will be necessary and recognize compiler vs. runtime errors related to object reference casting.
- Explain the effect of modifiers on inheritance with respect to constructors, instance or static variables, and instance or static methods.
- Given a scenario, develop code that declares and/or invokes overridden or overloaded methods and code that declares and/or invokes superclass, or overloaded constructors.
- Develop code that implements "is-a" and/or "has-a" relationships.

Section 6: Collections / Generics

- Given a design scenario, determine which collection classes and/or interfaces should be used to properly implement that design, including the use of the Comparable interface.
- Distinguish between correct and incorrect overrides of corresponding hashCode and equals methods, and explain the difference between == and the equals method.
- Write code that uses the generic versions of the Collections API, in particular, the Set, List, and Map interfaces and implementation classes. Recognize the limitations of the non-generic Collections API and how to refactor code to use the generic versions. Write code that uses the NavigableSet and NavigableMap interfaces.
- Develop code that makes proper use of type parameters in class/interface declarations, instance variables, method arguments, and return types; and write generic methods or methods that make use of wildcard types and understand the similarities and differences between these two approaches.
- Use capabilities in the java.util package to write code to manipulate a list by sorting, performing a binary search, or converting the list to an array. Use capabilities in the java.util package to write code to manipulate an array by sorting, performing a binary search, or converting the array to a list. Use the java.util.Comparator and java.lang.Comparable interfaces to affect the sorting of lists and arrays. Furthermore, recognize the effect of the "natural ordering" of primitive wrapper classes and java.lang.String on sorting.

Section 7: Fundamentals

- Given a code example and a scenario, write code that uses the appropriate access modifiers, package declarations, and import statements to interact with (through access or inheritance) the code in the example.
- Given an example of a class and a command-line, determine the expected runtime behavior.
- Determine the effect upon object references and primitive values when they are passed into methods that perform assignments or other modifying operations on the parameters.
- Given a code example, recognize the point at which an object becomes eligible for garbage collection, determine what is and is not guaranteed by the garbage collection system, and recognize the behaviors of the Object.finalize() method.
- Given the fully-qualified name of a class that is deployed inside and/or outside a JAR file, construct the appropriate directory structure for that class. Given a code example and a classpath, determine whether the classpath will allow the code to compile successfully.
- Write code that correctly applies the appropriate operators including assignment operators (limited to: =, +=, -=), arithmetic operators (limited to: +, -, *, /, %, ++, --), relational operators (limited to: <, <=, >, >=, ==, !=), the instanceof operator, logical operators (limited to: &, |, ^, !, &&, ||), and the conditional operator (?:), to produce a desired result. Write code that determines the equality of two objects or two primitives

Exam A

QUESTION 1

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());

Correct Answer: B

QUESTION 2

Given:

```
1. public class KungFu {
2.     public static void main(String[] args) {
3.         Integer x = 400;
4.         Integer y = x;
5.         x++;
6.         StringBuilder sb1 = new StringBuilder("123");
7.         StringBuilder sb2 = sb1;
8.         sb1.append("5");
9.         System.out.println((x == y) + " " + (sb1 == sb2));
10.    }
11. }
```

What is the result?

- A. true true
- B. false true
- C. true false
- D. false false
- E. Compilation fails.
- F. An exception is thrown at runtime.

Correct Answer: B

QUESTION 3

Given:

```
11. class Converter {
12.     public static void main(String[] args) {
13.         Integer i = args[0];
14.         int j = 12;
```

```

15.         System.out.println("It is " + (j == i) + " that j==i.");
16.     }
17. }

```

What is the result when the programmer attempts to compile the code and run it with the command `java Converter 12`?

- A. It is true that `j==i`.
- B. It is false that `j==i`.
- C. An exception is thrown at runtime.
- D. Compilation fails because of an error in line 13.

Correct Answer: D

QUESTION 4

Given

```

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. }

```

and:

```

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

```

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.

Correct Answer: A

QUESTION 5

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; }`

Correct Answer: AB

QUESTION 6

Given:

```

1. public class Plant {
2.     private String name;
3.
4.     public Plant(String name) {
5.         this.name = name;

```

```

6.     }
7.
8.     public String getName() {
9.         return name;
10.    }
11.}

1. public class Tree extends Plant {
2.     public void growFruit() {
3.     }
4.
5.     public void dropLeaves() {
6.     }
7. }

```

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.

Correct Answer: D

QUESTION 7

Click the Exhibit button.

```

1. public class GoTest {
2.     public static void main(String[] args) {
3.         Sente a = new Sente(); a.go();
4.         Goban b = new Goban(); b.go();
5.         Stone c = new Stone(); c.go();
6.     }
7. }
8.
9. class Sente implements Go {
10.    public void go(){
11.        System.out.println("go in Sente");
12.    }
13. }
14.
15. class Goban extends Sente {
16.    public void go(){
17.        System.out.println("go in Goban");
18.    }
19. }
20. }
21. class Stone extends Goban implements Go{
22. }
23.
24. interface Go { public void go(); }

```

What is the result?

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

Correct Answer: C

QUESTION 8

Given:

```
public interface A111 {  
    String s = "yo";  
  
    public void method1();  
}  
  
interface B {  
}  
  
interface C extends A111, B {  
    public void method1();  
  
    public void method1(int x);  
}
```

What is the result?

- A. Compilation succeeds.
- B. Compilation fails due to multiple errors.
- C. Compilation fails due to an error only on line 20.
- D. Compilation fails due to an error only on line 21.
- E. Compilation fails due to an error only on line 22.
- F. Compilation fails due to an error only on line 12.

Correct Answer: A

QUESTION 9

Click the Exhibit button.

```
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[] args) {  
32.         new Beta().testFoo();  
33.     }  
34. }
```

Which three statements are true? (Choose three.)

- 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.

Correct Answer: BEF

QUESTION 10

Given:

```
class Alpha {
    public void foo() { System.out.print("Afoo "); }
}
public class Beta extends Alpha {
    public void foo() { System.out.print("Bfoo "); }
    public static void main(String[] args) {
        Alpha a = new Beta();
        Beta b = (Beta)a;
        a.foo();
        b.foo();
    }
}
```

What is the result?

- A. Afoo Afoo
- B. Afoo Bfoo
- C. Bfoo Afoo
- D. Bfoo Bfoo
- E. Compilation fails.
- F. An exception is thrown at runtime.

Correct Answer: D

QUESTION 11

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.

Correct Answer: C

QUESTION 12

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".

Correct Answer: E

QUESTION 13

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

Correct Answer: ACE

QUESTION 14

Given that Triangle implements Runnable, and:

```

void go() throws Exception {

```



```

Thread t = new Thread(new Triangle());
t.start();
for(int x = 1; x < 100000; x++) {
    //insert code here line 35
    if(x%100 == 0) System.out.print("g");
}

public void run() {
    try {
        for(int x = 1; x < 100000; x++) {
            // insert the same code here line 41
            if(x%100 == 0) System.out.print("t");
        }
    } catch (Exception e) {
    }
}
}

```

Which two statements, inserted independently at both lines 35 and 41, tend to allow both threads to temporarily pause and allow the other thread to execute? (Choose two.)

- A. Thread.wait();
- B. Thread.join();
- C. Thread.yield();
- D. Thread.sleep(1);
- E. Thread.notify();

Correct Answer: CD

QUESTION 15

Given:

```

public class Yikes {

    public static void go(Long n) {
        System.out.print("Long ");
    }

    public static void go(Short n) {
        System.out.print("Short ");
    }

    public static void go(int n) {
        System.out.print("int ");
    }

    public static void main(String[] args) {
        short y = 6;
        long z = 7;
        go(y);
        go(z);
    }
}

```

What is the result?

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

Correct Answer: A

QUESTION 16

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-2000.
- 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.

Correct Answer: D

QUESTION 17

Which two scenarios are NOT safe to replace a StringBuffer object with a StringBuilder object? (Choose two.)

- A. When using versions of Java technology earlier than 5.0.
- B. When sharing a StringBuffer among multiple threads.
- C. When using the java.io class StringBufferInputStream.
- D. When you plan to reuse the StringBuffer to build more than one string.

Correct Answer: AB

QUESTION 18

Given that c is a reference to a valid java.io.Console object, and:

```
11. String pw = c.readPassword("%s", "pw: ");
12. System.out.println("got " + pw);
13. String name = c.readLine("%s", "name: ");
14. System.out.println(" got ", name);
```

If the user types fido when prompted for a password, and then responds bob when prompted for a name, what is the result?

- A. pw: got fido name: bob got bob
- B. pw: fido got fido name: bob got bob
- C. pw: got fido name: bob got bob
- D. pw: fido got fido name: bob got bob
- E. Compilation fails.
- F. An exception is thrown at runtime.

Correct Answer: E

QUESTION 19

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.

Correct Answer: D

QUESTION 20

Given:

```
import java.io.*;

class Animal {
    Animal() {
        System.out.print("a");
    }
}

class Dog extends Animal implements Serializable {
    Dog() {
        System.out.print("d");
    }
}

public class Beagle extends Dog {
}
```

If an instance of class Beagle is created, then Serialized, then deSerialized, what is the result?

- A. ad
- B. ada
- C. add
- D. adad
- E. Compilation fails.
- F. An exception is thrown at runtime.

Correct Answer: B

QUESTION 21

A team of programmers is involved in reviewing a proposed design for a new utility class. After some discussion, they realize that the current design allows other classes to access methods in the utility class that should be accessible only to methods within the utility class itself. What design issue has the team discovered?

- A. Tight coupling
- B. Low cohesion
- C. High cohesion
- D. Loose coupling
- E. Weak encapsulation
- F. Strong encapsulation

Correct Answer: E

QUESTION 22

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"); }`

Correct Answer: D

QUESTION 23

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;`

Correct Answer: D

QUESTION 24

Given:

```
class Animal {
    public String noise() {
        return "peep";
    }
}
```

```
class Dog extends Animal {
    public String noise() {
        return "bark";
    }
}
```

```
class Cat extends Animal {
    public String noise() {
        return "meow";
    }
}
```

```
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.

Correct Answer: E

QUESTION 25

Given:

```
abstract class A {  
    abstract void a1();  
  
    void a2() {  
    }  
}  
  
class B extends A {  
    void a1() {  
    }  
  
    void a2() {  
    }  
}  
  
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();

Correct Answer: ABDF

QUESTION 26

Given:

```
class Employee {  
    String name;  
    double baseSalary;  
  
    Employee(String name, double baseSalary) {  
        this.name = name;  
        this.baseSalary = baseSalary;  
    }  
}
```

```

09. public class SalesPerson extends Employee {
10.     double commission;
11.
12.     public SalesPerson(String name, double baseSalary, double commission) {
13.         // insert code here
14.     }
15. }

```

Which two code fragments, inserted independently at line 13, will compile? (Choose two.)

- A. `super(name, baseSalary);`
- B. `this.commission = commission;`
- C. `super(); this.commission = commission;`
- D. `this.commission = commission; super();`
- E. `super(name, baseSalary); this.commission = commission;`
- F. `this.commission = commission; super(name, baseSalary);`
- G. `super(name, baseSalary, commission);`

Correct Answer: AE

QUESTION 27

Given that: Gadget has-a Sprocket and Gadget has-a Spring and Gadget is-a Widget and Widget has-a Sprocket Which two code fragments represent these relationships? (Choose two.)

- A.

```
class Widget {
    Sprocket s;
}

class Gadget extends Widget {
    Spring s;
}
```
- B.

```
class Widget {
}

class Gadget extends Widget {
    Spring s1;
    Sprocket s2;
}
```
- C.

```
class Widget {
    Sprocket s1;
    Spring s2;
}

class Gadget extends Widget {
}
```
- D.

```
class Gadget {
    Spring s;
}

class Widget extends Gadget {
    Sprocket s;
}
```
- E.

```
class Gadget {
}

class Widget extends Gadget {
    Sprocket s1;
    Spring s2;
}
```
- F.

```
class Gadget {
    Spring s1;
    Sprocket s2;
}
```

```
class Widget extends Gadget {  
}
```

Correct Answer: AC

QUESTION 28

Given:

```
class Pizza {  
    java.util.ArrayList toppings;  
  
    public final void addTopping(String topping) {  
        toppings.add(topping);  
    }  
    public void removeTopping(String topping) {  
        toppings.remove(topping);  
    }  
}  
  
public class PepperoniPizza extends Pizza {  
    public void addTopping(String topping) {  
        System.out.println("Cannot add Toppings");  
    }  
  
    public static void main(String[] args) {  
        Pizza pizza = new PepperoniPizza();  
        pizza.addTopping("Mushrooms");  
        pizza.removeTopping("Peperoni");  
    }  
}
```

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.

Correct Answer: A

QUESTION 29

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 X only if the subclass is in the same package as X.

Correct Answer: BEF

QUESTION 30

Click the Exhibit button.

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. }

```

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

- A. insert a call to this() in the Car constructor
- B. insert a call to this() in the MeGo constructor
- C. insert a call to super() in the MeGo constructor
- D. insert a call to super(vin) in the MeGo constructor
- E. change the wheelCount variable in Car to protected
- F. change line 3 in the MeGo class to super.wheelCount = 3;

Correct Answer: DE

QUESTION 31

Click the Exhibit button.

```

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

```

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

- 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.

Correct Answer: D

QUESTION 32

Given:

```
public class Person {
    private String name, comment;
    private int age;

    public Person(String n, int a, String c) {
        name = n;
        age = a;
        comment = c;
    }

    public boolean equals(Object o) {
        if (!(o instanceof Person))
            return false;
        Person p = (Person) o;
        return age == p.age && name.equals(p.name);
    }
}
```

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;

Correct Answer: B

QUESTION 33

Given:

```
public class Key {
    private long id1;
    private long id2;

    // class Key methods
}
```

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)

Correct Answer: AD

QUESTION 34

Given:

```
03. import java.util.*;
04. public class Hancock {
05.     // insert code here
06.     list.add("foo");
07. }
08. }
```

Which two code fragments, inserted independently at line 5, will compile without warnings? (Choose two.)

- A. `public void addStrings(List list) {`
- B. `public void addStrings(List<String> list) {`
- C. `public void addStrings(List<? super String> list) {`
- D. `public void addStrings(List<? extends String> list) {`

Correct Answer: BC

QUESTION 35

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`

Correct Answer: D

QUESTION 36

Given a class whose instances, when found in a collection of objects, are sorted by using the `compareTo()` method, which two statements are true? (Choose two.)

- A. The class implements `java.lang.Comparable`.
- B. The class implements `java.util.Comparator`.
- C. The interface used to implement sorting allows this class to define only one sort sequence.
- D. The interface used to implement sorting allows this class to define many different sort sequences.

Correct Answer: AC

QUESTION 37

Given:

```
1. import java.util.*;
2.
3. public class Explorer3 {
4.     public static void main(String[] args) {
5.         TreeSet<Integer> s = new TreeSet<Integer>();
6.         TreeSet<Integer> subs = new TreeSet<Integer>();
7.         for (int i = 606; i < 613; i++)
8.             if (i % 2 == 0)
9.                 s.add(i);
10.        subs = (TreeSet) s.subSet(608, true, 611, true);
11.        subs.add(629);
12.        System.out.println(s + " " + subs);
13.    }
14. }
```

What is the result?

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

Correct Answer: B

QUESTION 38

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) {
12.            System.out.print(s + " ");
13.        }
14.    }
15. }
```

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.

Correct Answer: C

QUESTION 39

Given:

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

Which command-line invocations will compile?

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

Correct Answer: B

QUESTION 40

Click the Exhibit button

```
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. }

```

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.)

- 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.

Correct Answer: BE

QUESTION 41

Given:

```

1. public class Venus {
2.     public static void main(String[] args) {
3.         int[] x = { 1, 2, 3 };
4.         int y[] = { 4, 5, 6 };
5.         new Venus().go(x, y);
6.     }
7.
8.     void go(int[]... z) {
9.         for (int[] a : z)
10.            System.out.print(a[0]);
11.     }
12. }

```

What is the result?

- A. 1
- B. 12
- C. 14
- D. 123

- E. Compilation fails.
- F. An exception is thrown at runtime.

Correct Answer: C

QUESTION 42

Given:

```
1. public class Test {
2.     public enum Dogs {collie, harrier, shepherd};
3.     public static void main(String [] args) {
4.         Dogs myDog = Dogs.shepherd;
5.         switch (myDog) {
6.             case collie:
7.                 System.out.print("collie ");
8.             case default:
9.                 System.out.print("retriever ");
10.            case harrier:
11.                System.out.print("harrier ");
12.        }
13.    }
14. }
```

What is the result?

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

Correct Answer: D

QUESTION 43

Given:

```
1. public class Breaker2 {
2.     static String o = "";
3.
4.     public static void main(String[] args) {
5.         z: for (int x = 2; x < 7; x++) {
6.             if (x == 3)
7.                 continue;
8.             if (x == 5)
9.                 break z;
10.            o = o + x;
11.        }
12.        System.out.println(o);
13.    }
14. }
```

What is the result?

- A. 2
- B. 24
- C. 234
- D. 246
- E. 2346

F. Compilation fails.

Correct Answer: B

QUESTION 44

Given:

```
public static void main(String[] args) {
    String str = "null";
    if (str == null) {
        System.out.println("null");
    } else (str.length() == 0) {
        System.out.println("zero");
    } else {
        System.out.println("some");
    }
}
```

What is the result?

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

Correct Answer: D

QUESTION 45

Given:

```
import java.io.IOException;
class A {
    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.
20.     public static void main(String[] args) {
21.         try {
22.             new B().process();
23.         } catch (IOException e) {
24.             System.out.println("Exception");
25.         }
26.     }
27. }
```

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.

Correct Answer: D

QUESTION 46

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.

Correct Answer: D

QUESTION 47

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

Correct Answer: D

QUESTION 48

Given:

```
public class Spock {
    public static void main(String[] args) {
        Long tail = 2000L;
        Long distance = 1999L;
```

```

        Long story = 1000L;
        if ((tail > distance) ^ ((story * 2) == tail))
            System.out.print("1");
        if ((distance + 1 != tail) ^ ((story * 2) == distance))
            System.out.print("2");
    }
}

```

What is the result?

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

Correct Answer: E

QUESTION 49

Given:

```

1. interface DeclareStuff {
2.     public static final int EASY = 3;
3.
4.     void doStuff(int t);
5. }
6.
7. public class TestDeclare implements DeclareStuff {
8.     public static void main(String[] args) {
9.         int x = 5;
10.        new TestDeclare().doStuff(++x);
11.    }
12.
13.    void doStuff(int s) {
14.        s += EASY + ++s;
15.        System.out.println("s " + s);
16.    }
17.}

```

What is the result?

- A. s 14
- B. s 16
- C. s 10
- D. Compilation fails.
- E. An exception is thrown at runtime.

Correct Answer: D

QUESTION 50

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

Correct Answer: C

QUESTION 51

Given:

```
01. import java.io.*;
02.
03. public class Talk {
04.     public static void main(String[] args) {
05.         Console c = new Console();
06.         String pw;
07.         System.out.print("password: ");
08.         pw = c.readLine();
09.         System.out.println("got " + pw);
10.     }
11. }
```

If the user types the password aiko when prompted, what is the result?

- A. password:
got
- B. password:
got aiko
- C. password: aiko
got aiko
- D. An exception is thrown at runtime.
- E. Compilation fails due to an error on line 5.

Correct Answer: E

QUESTION 52

Given the following code:

```
package console;

public class Ques02 {

    public static void main(String[] args) {

        int anInt = 100;
        double aDouble = 100.00;

        System.out.format("%2d - %1f", anInt, aDouble);
    }
}
```

What is the output?

- A. The program will output '100.000000 – 100'.
- B. The program will output '100 - 100.000000'.
- C. The program will throw a `IllegalFormatException` at runtime.
- D. The program will output '100 – 100'.

Correct Answer: B

QUESTION 53

Given:

```
import java.util.*;

public class Values {

    public static void main(String[] args) {

        Properties p = System.getProperties();
        p.setProperty("myProp", "myValue");
        System.out.print(p.getProperty("cmdProp") + " ");
        System.out.print(p.getProperty("myProp") + " ");
        System.out.print(p.getProperty("noProp") + " ");
        p.setProperty("cmdProp", "newValue");
        System.out.print(p.getProperty("cmdProp"));

    }
}
```

And given the command line invocation:

```
java -DcmdProp=cmdValue Values
```

- A. null myValue null null
- B. cmdValue null null cmdValue
- C. cmdValue null null newValue
- D. cmdValue myValue null cmdValue
- E. cmdValue myValue null newValue
- F. An exception is thrown at runtime

Correct Answer: E

QUESTION 54

Given this method in a class:

```
public String toString() {
    StringBuffer buffer = new StringBuffer();
    buffer.append('<');
    buffer.append(this.name);
    buffer.append('>');
    return buffer.toString();
}
```

Which 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 well and converting the code to use StringBuilder will not enhance the performance
- D. This code will perform poorly. For better performance, the code should be rewritten: return "<"+ this.name + ">";

Correct Answer: B

QUESTION 55

Given a class Repetition:

```
package utils;

public class Repetition {
    public static String twice(String s) { return s + s; }
}
```

and given another class Demo:

```

01. public class Demo {
02.     public static void main(String[] args) {
03.         System.out.println(twice("pizza"));
04.     }
05. }

```

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;

Correct Answer: F

QUESTION 56

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.

Correct Answer: A

QUESTION 57

Given:

```

1. package sun.scjp;
2. public enum Color { RED, GREEN, BLUE }

1. package sun.beta;
2. // insert code here
3. public class Beta {
4.     Color g = GREEN;
5.     public static void main( String[] argv)
6.     { System.out.println(GREEN); }
7. }

```

The class Beta and the enum Color are in different packages.

Which two code fragments, inserted individually at line 2 of the Beta declaration, will allow this code to compile? (Choose two.)

- A. import sun.scjp.Color.*;
- B. import static sun.scjp.Color.*;
- C. import sun.scjp.Color;
 - import static sun.scjp.Color.*;
- D. import sun.scjp.*;
 - import sun.scjp.Color.*;

E. `import sun.scjp.Color;`
`import static sun.scjp.Color.GREEN;`

Correct Answer: CE

QUESTION 58

Given:

```
1. interface DeclareStuff {
2.     public static final int EASY = 3;
3.
4.     void doStuff(int t);
5. }
6.
7. public class TestDeclare implements DeclareStuff {
8.     public static void main(String[] args) {
9.         int x = 5;
10.        new TestDeclare().doStuff(++x);
11.    }
12.
13.    void doStuff(int s) {
14.        s += EASY + ++s;
15.        System.out.println("s " + s);
16.    }
17. }
```

What is the result?

- A. s 14
- B. s 16
- C. s 10
- D. Compilation fails.
- E. An exception is thrown at runtime.

Correct Answer: D

QUESTION 59

Given:

```
11. class X { public void foo() { System.out.print("X "); } }
12.
13. public class SubB extends X {
14.     public void foo() throws RuntimeException {
15.         super.foo();
16.         if (true) throw new RuntimeException();
17.         System.out.print("B ");
18.     }
19.     public static void main(String[] args) {
20.         new SubB().foo();
21.     }
22. }
```

What is the result?

- A. X, followed by an Exception.
- B. No output, and an Exception is thrown.
- C. Compilation fails due to an error on line 14.
- D. Compilation fails due to an error on line 16.
- E. Compilation fails due to an error on line 17.
- F. X, followed by an Exception, followed by B.

Correct Answer: A

QUESTION 60

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.

Correct Answer: A