

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

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:

```
01. interface Animal { void makeNoise(); }
02. class Horse implements Animal {
03.     Long weight = 1200L;
04.     public void makeNoise() { System.out.println("whinny"); }
05. }
06.
07. public class Icelandic extends Horse {
08.     public void makeNoise() { System.out.println("vinny"); }
09.     public static void main(String[] args) {
10.         Icelandic i1 = new Icelandic();
11.         Icelandic i2 = new Icelandic();
12.         Icelandic i3 = new Icelandic();
13.         i3 = i1; i1 = i2; i2 = null; i3 = i1;
14.     }
15. }
```

When line 14 is reached, how many objects are eligible for the garbage collector?

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4
- F. 6

Correct Answer: E

QUESTION 2

Given two files, GrizzlyBear.java and Salmon.java:

```
01. package animals.mammals;
02.
03. public class GrizzlyBear extends Bear {
04.     void hunt() {
05.         Salmon s = findSalmon();
06.         s.consume();
07.     }
08. }

01. package animals.fish;
02.
03. public class Salmon extends Fish {
04.     public void consume() { /* do stuff */ }
05. }
```

If both classes are in the correct directories for their packages, and the Mammal class correctly defines the findSalmon() method, which change allows this code to compile?

- A. add import animals.mammals.*; at line 2 in Salmon.java
- B. add import animals.fish.*; at line 2 in GrizzlyBear.java
- C. add import animals.fish.Salmon.*; at line 2 in GrizzlyBear.java
- D. add import animals.mammals.GrizzlyBear.*; at line 2 in Salmon.java

Correct Answer: B

QUESTION 3

Given:

```
String[] elements = { "for", "tea", "too" };
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].

Correct Answer: D

QUESTION 4

A company that makes Computer Assisted Design (CAD) software has, within its application, some utility classes that are used to perform 3D rendering tasks. The company's chief scientist has just improved the performance of one of the utility classes' key rendering algorithms, and has assigned a programmer to replace the old algorithm with the new algorithm. When the programmer begins researching the utility classes, she is happy to discover that the algorithm to be replaced exists in only one class. The programmer reviews that class's API, and replaces the old algorithm with the new algorithm, being careful that her changes adhere strictly to the class's API. Once testing has begun, the programmer discovers that other classes that use the class she changed are no longer working properly. What design flaw is most likely the cause of these new bugs?

- A. Inheritance
- B. Tight coupling
- C. Low cohesion
- D. High cohesion
- E. Loose coupling
- F. Object immutability

Correct Answer: B

QUESTION 5

Given:

```
class ClassA {
    public int numberOfInstances;

    protected ClassA(int numberOfInstances) {
        this.numberOfInstances = numberOfInstances;
    }
}

public class ExtendedA extends ClassA {
    private ExtendedA(int numberOfInstances) {
        super(numberOfInstances);
    }

    public static void main(String[] args) {
        ExtendedA ext = new ExtendedA(420);
        System.out.print(ext.numberOfInstances);
    }
}
```

Which statement is true?

- A. 420 is the output.
- B. An exception is thrown at runtime.
- C. All constructors must be declared public.
- D. Constructors CANNOT use the private modifier.

E. Constructors CANNOT use the protected modifier.

Correct Answer: A

QUESTION 6

Given:

```
class ClassA {}  
class ClassB extends ClassA {}  
class ClassC extends ClassA {}
```

and:

```
ClassA p0 = new ClassA();  
ClassB p1 = new ClassB();  
ClassC p2 = new ClassC();  
ClassA p3 = new ClassB();  
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;

Correct Answer: AEF

QUESTION 7

Given:

```
class Thingy { Meter m = new Meter(); }  
  
class Component { void go() { System.out.print("c"); } }  
  
class Meter extends Component { void go() { System.out.print("m"); } }  
  
class DeluxeThingy extends Thingy {  
    public static void main(String[] args) {  
        DeluxeThingy dt = new DeluxeThingy();  
        dt.m.go();  
        Thingy t = new DeluxeThingy();  
        t.m.go();  
    }  
}
```

Which two are true? (Choose two.)

- A. The output is mm.
- B. The output is mc.
- C. Component is-a Meter.
- D. Component has-a Meter.
- E. DeluxeThingy is-a Component.
- F. DeluxeThingy has-a Component.

Correct Answer: AF

QUESTION 8

Given:

```

interface Jumper { public void jump(); }

class Animal {}

class Dog extends Animal {
    Tail tail;
}

class Beagle extends Dog implements Jumper{
    public void jump() {}
}

class Cat implements Jumper{
    public void jump() {}
}

```

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

Correct Answer: BCF

QUESTION 9

Given:

```

import java.util.*;

public class WrappedString {
    private String s;

    public WrappedString(String s) { this.s = s; }

    public static void main(String[] args) {
        HashSet<Object> hs = new HashSet<Object>();
        WrappedString ws1 = new WrappedString("aardvark");
        WrappedString ws2 = new WrappedString("aardvark");
        String s1 = new String("aardvark");
        String s2 = new String("aardvark");
        hs.add(ws1); hs.add(ws2); hs.add(s1); hs.add(s2);
        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.

Correct Answer: D

QUESTION 10

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> {

Correct Answer: DF

QUESTION 11

Given:

```
3. import java.util.*;
4. public class G1 {
5.     public void takeList(List<? extends String> list) {
6.         // insert code here
7.     }
8. }
```

Which three code fragments, inserted independently at line 6, will compile? (Choose three.)

- A. list.add("foo");
- B. Object o = list;
- C. String s = list.get(0);
- D. list = new ArrayList<String>();
- E. list = new ArrayList<Object>();

Correct Answer: BCD

QUESTION 12

Given that the elements of a PriorityQueue are ordered according to natural ordering, and:

```
import java.util.*;

public class GetInLine {
    public static void main(String[] args) {
        PriorityQueue<String> pq = new PriorityQueue<String>();
        pq.add("banana");
        pq.add("pear");
        pq.add("apple");
        System.out.println(pq.poll() + " " + pq.peek());
    }
}
```

What is the result?

- A. apple pear
- B. banana pear
- C. apple apple
- D. apple banana
- E. banana banana

Correct Answer: D

QUESTION 13

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 14

Given:

```
import java.util.*;

public class Mapit {
    public static void main(String[] args) {
        Set<Integer> set = new HashSet<Integer>();
        Integer i1 = 45;
        Integer i2 = 46;
        set.add(i1);
        set.add(i1);
        set.add(i2); System.out.print(set.size() + " ");
        set.remove(i1); System.out.print(set.size() + " ");
        i2 = 47;
        set.remove(i2); System.out.print(set.size() + " ");
    }
}
```

What is the result?

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

Correct Answer: B

QUESTION 15

Given:

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



```

public class Explorer1 {
    public static void main(String[] args) {
        TreeSet<Integer> s = new TreeSet<Integer>();
        TreeSet<Integer> subs = new TreeSet<Integer>();
        for(int i = 606; i < 613; i++)
            if(i%2 == 0) s.add(i);
        subs = (TreeSet)s.subSet(608, true, 611, true);
        s.add(609);
        System.out.println(s + " " + subs);
    }
}

```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. [608, 609, 610, 612] [608, 610]
- D. [608, 609, 610, 612] [608, 609, 610]
- E. [606, 608, 609, 610, 612] [608, 610]
- F. [606, 608, 609, 610, 612] [608, 609, 610]

Correct Answer: F

QUESTION 16

Given:

```

import java.util.*;

public class Quest {
    public static void main(String[] args) {
        String[] colors = {"blue", "red", "green", "yellow", "orange"};
        Arrays.sort(colors);
        int s2 = Arrays.binarySearch(colors, "orange");
        int s3 = Arrays.binarySearch(colors, "violet");
        System.out.println(s2 + " " + s3);
    }
}

```

What is the result?

- A. 2 -1
- B. 2 -4
- C. 2 -5
- D. 3 -1
- E. 3 -4
- F. 3 -5
- G. Compilation fails.
- H. An exception is thrown at runtime.

Correct Answer: C

QUESTION 17

Given:

```

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);`

Correct Answer: B

QUESTION 18

Which two statements are true? (Choose two.)

- A. It is possible to synchronize static methods.
- B. When a thread has yielded as a result of `yield()`, it releases its locks.
- C. When a thread is sleeping as a result of `sleep()`, it releases its locks.
- D. The `Object.wait()` method can be invoked only from a synchronized context.
- E. The `Thread.sleep()` method can be invoked only from a synchronized context.
- F. When the thread scheduler receives a `notify()` request, and notifies a thread, that thread immediately releases its lock.

Correct Answer: AD

QUESTION 19

Given:

```
public class TestOne implements Runnable {
    public static void main (String[] args) throws Exception {
        Thread t = new Thread(new TestOne());
        t.start();
        System.out.print("Started");
        t.join();
        System.out.print("Complete");
    }

    public void run() {
        for (int i = 0; i < 4; i++) {
            System.out.print(i);
        }
    }
}
```

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".

Correct Answer: E

QUESTION 20

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 */ }
}

```

Correct Answer: CEF

QUESTION 21

Given:

```

1. public class TestFive {
2.     private int x;
3.
4.     public void foo() {
5.         int current = x;
6.         x = current + 1;
7.     }
8.
9.     public void go() {
10.        for(int i = 0; i < 5; i++) {
11.            new Thread() {
12.                public void run() {
13.                    foo();
14.                    System.out.print(x + ", ");
15.                }
16.            }.start();
17.        }
18.    }
19.}

```

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

- A. move the line 14 print statement into the foo() method
- B. change line 9 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 }

Correct Answer: AD

QUESTION 22

Given that t1 is a reference to a live thread, which is true?

- A. The Thread.sleep() method can take t1 as an argument.
- B. The Object.notify() method can take t1 as an argument.
- C. The Thread.yield() method can take t1 as an argument.
- D. The Thread.setPriority() method can take t1 as an argument.
- E. The Object.notify() method arbitrarily chooses which thread to notify.

Correct Answer: E

QUESTION 23

Given:

```

Runnable r = new Runnable() {
    public void run() {
        System.out.print("Cat");
    }
}

```

```

    }
};

Thread t = new Thread(r) {
    public void run() {
        System.out.print("Dog");
    }
};

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.

Correct Answer: B

QUESTION 24

Given:

```

public class Threads5 {
    public static void main (String[] args) {
        new Thread(new Runnable() {
            public void run() {
                System.out.print("bar");
            }).start();
        }
    }
}

```

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.

Correct Answer: C

QUESTION 25

Given:

```

09. class One {
10.     void foo() { }
11. }
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 */ }`

Correct Answer: BCE

QUESTION 26

Given:

```
08. abstract public class Employee {
09.     protected abstract double getSalesAmount();
10.
11.     public double getCommision() {
12.         return getSalesAmount() * 0.15;
13.     }
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; }

Correct Answer: BD

QUESTION 27

Given:

```
1. class X {
2.     X() { System.out.print(1); }
3.
4.     X(int x) {
5.         this(); System.out.print(2);
6.     }
7. }
8.
9. public class Y extends X {
10.     Y() { super(6); System.out.print(3); }
11.
12.     Y(int y) {
13.         this(); System.out.println(4);
14.     }
15.
16.     public static void main(String[] a) { new Y(5); }
17. }
```

What is the result?

- A. 13
- B. 134
- C. 1234
- D. 2134
- E. 2143
- F. 4321

Correct Answer: C

QUESTION 28

Given:

```
package com.sun.scjp;
```

```
public class Geodetics {
    public static final double DIAMETER = 12756.32; // kilometers
}
```

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; }
}
```

Correct Answer: AC

QUESTION 29

Given:

```
1. public class A {
2.     public void doit() {
3.     }
4.
5.     public String doit() {
6.         return "a";
7.     }
8.
9.     public double doit(int x) {
10.        return 1.0;
11.    }
12.}
```

What is the result?

- A. An exception is thrown at runtime.
- B. Compilation fails because of an error in line 9.
- C. Compilation fails because of an error in line 5.
- D. Compilation succeeds and no runtime errors with class A occur.

Correct Answer: C

QUESTION 30

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.

Correct Answer: AD

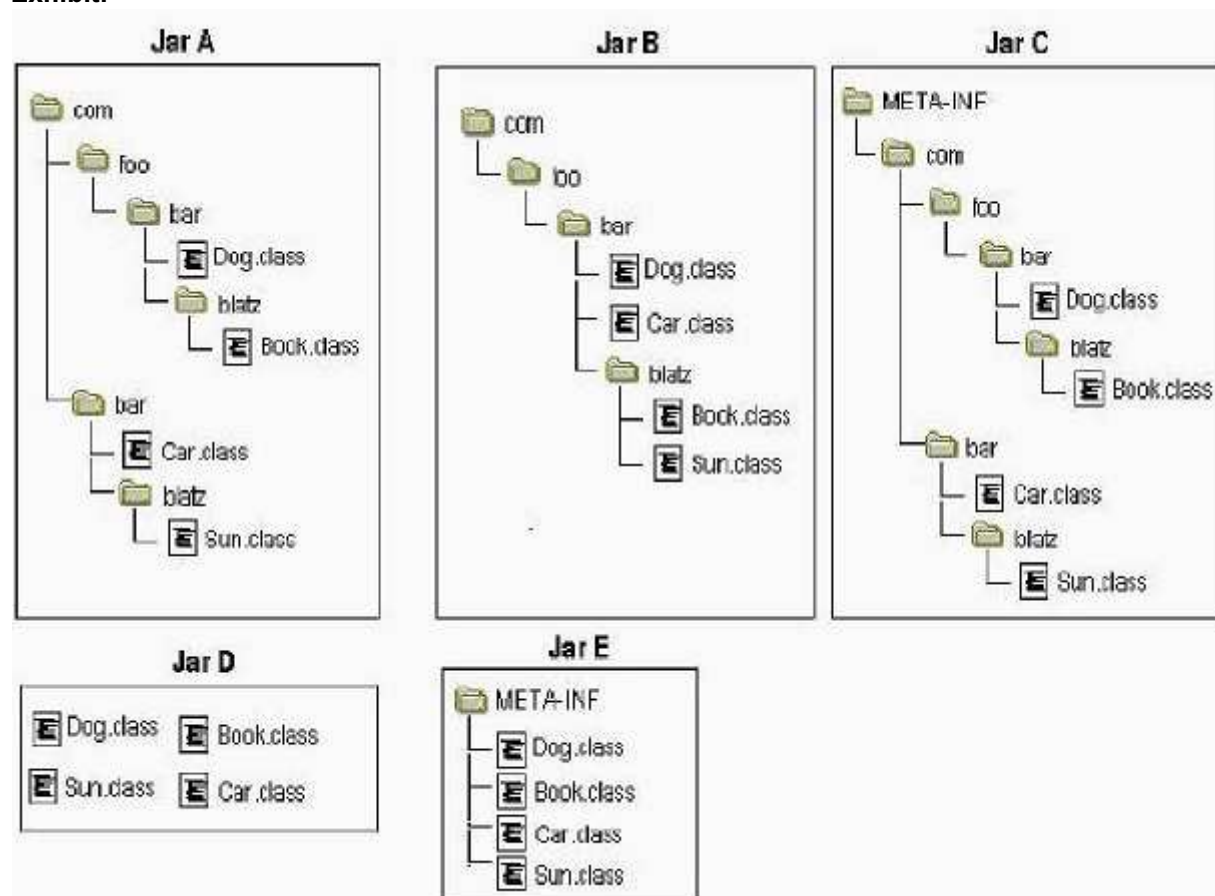
QUESTION 31

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?

Exhibit:



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

Correct Answer: A

QUESTION 32

Given:

```
public class ClassA {
    public void methodA() {
        ClassB classB = new ClassB();
        classB.getValue();
    }
}

class ClassB {
    public ClassC classC;

    public String getValue() {
        return classC.getValue();
    }
}

class ClassC {
    public String value;

    public String getValue() {
        value = "ClassB";
        return value;
    }
}
```

and:

```
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.

Correct Answer: D

QUESTION 33

Given:

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

Correct Answer: C

QUESTION 34

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.
20. public static void main(String[] args) {
21.     System.out.println(Title.MR.format("Doe", "John"));
22. }
```

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 21.

Correct Answer: A

QUESTION 35

Given the following six method names:

- addListener
- addMouseListener
- setMouseListener
- deleteMouseListener
- removeMouseListener
- registerMouseListener

How many of these method names follow JavaBean Listener naming rules?

- A. 1
- B. 2
- C. 3
- D. 4
- E. 5

Correct Answer: B

QUESTION 36

Given:

```
09. class Line {
10.     public static class Point {}
11. }
12.
13. class Triangle {
14.     public Triangle(){
15.         // insert code here
16.     }
17. }
```

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();`

Correct Answer: B

QUESTION 37

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`

Correct Answer: ABD

QUESTION 38

Given:

```
public class A{  
    private int counter = 0;  
  
    public static int getInstanceCount() {  
        return counter;  
    }  
  
    public A() {  
        counter++;  
    }  
}
```

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());
```

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.

Correct Answer: A

QUESTION 39

Given classes defined in two different files:

```

package util;

public class BitUtils {
    public static void process(byte[] b) { /* more code here */ }
}

1. package app;
2.
3. public class SomeApp {
4.     public static void main(String[] args) {
5.         byte[] bytes = new byte[256];
6.         // insert code here
7.     }
8. }

```

What is required at line 6 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);

Correct Answer: C

QUESTION 40

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

```

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. }

```

- 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 );
```

Correct Answer: BCF

QUESTION 41

Given:

```
class Snoochy {
    Boochy booch;

    public Snoochy() { booch = new Boochy(this); }
}

class Boochy {
    Snoochy snooch;

    public Boochy(Snoochy s) { snooch = s; }
}
```

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.

Correct Answer: E

QUESTION 42

Given:

```
04. class Payload {
05.     private int weight;
06.     public Payload (int w) { weight = w; }
07.     public void setWeight(int w) { weight = w; }
08.     public String toString() { return Integer.toString(weight); }
09. }
10.
11. public class TestPayload {
12.     static void changePayload(Payload p) { /* insert code */ }
13.     public static void main(String[] args) {
14.         Payload p = new Payload(200);
15.         p.setWeight(1024);
16.         changePayload(p);
17.         System.out.println("p is " + p);
18.     }
19. }
```

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

- A. `p.setWeight(420);`
- B. `p.changePayload(420);`

- C. `p = new Payload(420);`
- D. `Payload.setWeight(420);`
- E. `p = Payload.setWeight(420);`

Correct Answer: A

QUESTION 43

Given:

```
public static void test(String str) {  
    int check = 4;  
    if (check = str.length()) {  
        System.out.print(str.charAt(check -= 1) + ", ");  
    } else {  
        System.out.print(str.charAt(0) + ", ");  
    }  
}
```

and the invocation:

```
test("four");  
test("tee");  
test("to");
```

What is the result?

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

Correct Answer: C

QUESTION 44

Given classes defined in two different files:

```
package util;  
public class BitUtils {  
    private static void process(byte[] b) {}  
}  
  
01. package app;  
02. public class SomeApp {  
03.     public static void main(String[] args) {  
04.         byte[] bytes = new byte[256];  
05.         // insert code here  
06.     }  
07. }
```

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`.

Correct Answer: F

QUESTION 45

Given:

```
public class Pass2 {
    public void main(String [] args) {
        int x = 6;
        Pass2 p = new Pass2();
        p.doStuff(x);
        System.out.print(" main x = " + x);
    }

    void doStuff(int x) {
        System.out.print(" doStuff x = " + x++);
    }
}
```

And the command-line invocations:

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

What is the result?

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

Correct Answer: B

QUESTION 46

Given:

```
public class Test {
    public enum Dogs {collie, harrier};

    public static void main(String [] args) {
        Dogs myDog = Dogs.collie;
        switch (myDog) {
            case collie:
                System.out.print("collie ");
            case harrier:
                System.out.print("harrier ");
        }
    }
}
```

What is the result?

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

Correct Answer: D

QUESTION 47

Given:

```
public class Donkey {
```

```

    public static void main(String[] args) {
        boolean assertsOn = false;
        assert (assertsOn) : assertsOn = true;
        if(assertsOn) {
            System.out.println("assert is on");
        }
    }
}

```

If class Donkey is invoked twice, the first time without assertions enabled, and the second time with assertions enabled, what are the results?

- A. no output
- B. no output
assert is on
- C. assert is on
- D. no output
An AssertionError is thrown.
- E. assert is on
An AssertionError is thrown.

Correct Answer: D

QUESTION 48

Given:

```

static void test() {
    try {
        String x = null;
        System.out.print(x.toString() + " ");
    }
    finally { System.out.print("finally "); }
}

public static void main(String[] args) {
    try { test(); }
    catch (Exception ex) { System.out.print("exception "); }
}

```

What is the result?

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

Correct Answer: E

QUESTION 49

Given:

```

static void test() throws Error {
    if (true) throw new AssertionError();
    System.out.print("test ");
}

public static void main(String[] args) {
    try { test(); }
    catch (Exception ex) { System.out.print("exception "); }
    System.out.print("end ");
}

```

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.

Correct Answer: E

QUESTION 50

Given:

```
01. class TestException extends Exception { }
02. class A {
03.     public String sayHello(String name) throws TestException {
04.         if(name == null) throw new TestException();
05.         return "Hello " + name;
06.     }
07. }
08. public class TestA {
09.     public static void main(String[] args) {
10.         new A().sayHello("Aiko");
11.     }
12. }
```

Which statement is true?

- A. Compilation succeeds.
- B. Class A does not compile.
- C. The method declared on line 9 cannot be modified to throw TestException.
- D. TestA compiles if line 10 is enclosed in a try/catch block that catches TestException.

Correct Answer: D

QUESTION 51

Given:

```
public static Collection get() {
    Collection sorted = new LinkedList();
    sorted.add("B"); sorted.add("C"); sorted.add("A");
    return sorted;
}

public static void main(String[] args) {
    for (Object obj: get()) {
        System.out.print(obj + ", ");
    }
}
```

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.

Correct Answer: B

QUESTION 52

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.

Correct Answer: A

QUESTION 53

Given:

```
public class Foo {
    static int[] a;
    static { a[0]=2; }
    public static void main( String[] args ) {}
}
```

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

Correct Answer: C

QUESTION 54

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.

Correct Answer: A

QUESTION 55

Given:

```
public static Iterator reverse(List list) {  
    Collections.reverse(list);  
    return list.iterator();  
}  
  
public static void main(String[] args) {  
    List list = new ArrayList();  
    list.add("1"); list.add("2"); list.add("3");  
    for (Object obj: reverse(list))  
        System.out.print(obj + ", ");  
}
```

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.

Correct Answer: C

QUESTION 56

Given:

```
01. public class TestString3 {  
02.     public static void main(String[] args) {  
03.         // insert code here  
04.         System.out.println(s);  
05.     }  
06. }
```

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");
delete(0,3).delete(1,3).delete(2,5).insert(1, "24");

Correct Answer: BE

QUESTION 57

Given:

- 1. d is a valid, non-null Date object
- 2. 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));

```

**Correct Answer: B**

### QUESTION 58

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

```

public class Starter extends Thread {
 private int x = 2;

 public static void main(String[] args) throws Exception {
 new Starter().makeItSo();
 }

 public Starter(){
 x = 5;
 start();
 }

 public void makeItSo() throws Exception {
 join();
 x = x - 1;
 System.out.println(x);
 }

 public void run() { x *= 2; }
}

```

- 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.

**Correct Answer: D**

### QUESTION 59

Given a correctly compiled class whose source code is:

```

1. package com.sun.sjcp;
2.
3. public class Commander {
4. public static void main(String[] args) {
5. // more code here
6. }
7. }

```

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`

**Correct Answer: B**

#### QUESTION 60

Given:

```
interface DoStuff2 {
 float getRange(int low, int high);
}

interface DoMore {
 float getAvg(int a, int b, int c);
}

abstract class DoAbstract implements DoStuff2, DoMore {
}

06. class DoStuff implements DoStuff2 {
07. public float getRange(int x, int y) {
08. return 3.14f;
09. }
10. }
11.
12. interface DoAll extends DoMore {
13. float getAvg(int a, int b, int c, int d);
14. }
```

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.

**Correct Answer: A**