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

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 programatically.

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/printf 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-quali fied 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:

```
public class Threads2 implements Runnable {
    public void run() {
        System.out.println("run.");
        throw new RuntimeException("Problem");
    }

    public static void main(String[] args) {
        Thread t = new Thread(new Threads2());
        t.start();
        System.out.println("End of method.");
    }
}
```

Which two can be results? (Choose two.)

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

Correct Answer: DE

QUESTION 2
Which two statements are true? (Choose two.)

A. It is possible for more than two threads to deadlock at once.
B. The JVM implementation guarantees that multiple threads cannot enter into a deadlocked state.
C. Deadlocked threads release once their sleep() method's sleep duration has expired.
D. Deadlocking can occur only when the wait(), notify(), and notifyAll() methods are used incorrectly.
E. It is possible for a single-threaded application to deadlock if synchronized blocks are used incorrectly.
F. If a piece of code is capable of deadlocking, you cannot eliminate the possibility of deadlocking by inserting invocations of Thread.yield().

Correct Answer: AF

QUESTION 3
Given:

```
void waitForSignal() {
    Object obj = new Object();
    synchronized (Thread.currentThread()) {
        obj.wait();
        obj.notify();
    }
}
```

Which statement is true?

A. This code can throw an InterruptedException.
B. This code can throw an IllegalMonitorStateException.

C. This code can throw a TimeoutException after ten minutes.
D. Reversing the order of obj.wait() and obj.notify() might cause this method to complete normally.
E. A call to notify() or notifyAll() from another thread might cause this method to complete normally.
F. This code does NOT compile unless "obj.wait()" is replaced with "((Thread) obj).wait()".

Correct Answer: B

QUESTION 4
Given:

```
class PingPong2 {
    synchronized void hit(long n) {
        for(int i = 1; i < 3; i++)
            System.out.print(n + "-" + i + " ");
    }
}

public class Tester implements Runnable {
    static PingPong2 pp2 = new PingPong2();

    public static void main(String[] args) {
        new Thread(new Tester()).start();
        new Thread(new Tester()).start();
    }

    public void run() { pp2.hit(Thread.currentThread().getId()); }
}
```

Which statement is true?

A. The output could be 5-1 6-1 6-2 5-2
B. The output could be 6-1 6-2 5-1 5-2
C. The output could be 6-1 5-2 6-2 5-1
D. The output could be 6-1 6-2 5-1 7-1

Correct Answer: B

QUESTION 5
Given:

```
public class Threads4 {
    public static void main (String[] args) {
        new Threads4().go();
    }

    public void go() {
        Runnable r = new Runnable() {
            public void run() {
                System.out.print("foo");
            }
        };
        Thread t = new Thread(r);
        t.start();
        t.start();
    }
}
```

What is the result?

A. Compilation fails.
B. An exception is thrown at runtime.
C. The code executes normally and prints "foo".

D. The code executes normally, but nothing is printed.

Correct Answer: B

QUESTION 6
Given:

```
public abstract class Shape {
    private int x;
    private int y;

    public abstract void draw();

    public void setAnchor(int x, int y) {
        this.x = x;
        this.y = y;
    }
}
```

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

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

Correct Answer: BE

QUESTION 7
Given:

```
public class Barn {
    public static void main(String[] args) {
        new Barn().go("hi", 1);
        new Barn().go("hi", "world", 2);
    }

    public void go(String... y, int x) {
        System.out.print(y[y.length - 1] + " ");
    }
}
```

What is the result?

A. hi hi

B. hi world
C. world world
D. Compilation fails.
E. An exception is thrown at runtime.

Correct Answer: D

QUESTION 8
Given:

```
class Nav{
    public enum Direction { NORTH, SOUTH, EAST, WEST }
}

public class Sprite{
    // insert code here
}
```

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

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

Correct Answer: D

QUESTION 9
Which statement is true about the classes and interfaces in the exhibit?

```
01. public interface A {
02.   public void doSomething(String thing);
03. }

01. public class AImpl implements A {
02.    public void doSomething(String msg) {}
03. }

01. public class B {
02.   public A doit(){
03.        //more code here
04.    }
05.    public String execute(){
06.       //more code here
07.    }
08. }

01. public class C extends B {
02.   public AImpl doit(){
03.        //more code here
04.    }
05.
06.    public Object execute() {
07.        //more code here
08.    }
09. }
```

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

Correct Answer: C

QUESTION 10
What is the result?

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

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

Correct Answer: D

QUESTION 11
Given:

```
01. public class Rainbow {
02.     public enum MyColor {
03.         RED(0xff0000), GREEN(0x00ff00), BLUE(0x0000ff);
04.         private final int rgb;
05.         MyColor(int rgb) { this.rgb = rgb; }
06.         public int getRGB() { return rgb; }
07.     };
08.     public static void main(String[] args) {
09.         //insert code here
10.     }
11. }
```

Which code fragment, inserted at line 9, allows the Rainbow class to compile?

A. `MyColor skyColor = BLUE;`
B. `MyColor treeColor = MyColor.GREEN;`
C. `if(RED.getRGB() < BLUE.getRGB()) { }`
D. Compilation fails due to other error(s) in the code.
E. `MyColor purple = new MyColor(0xff00ff);`
F. `MyColor purple = MyColor.BLUE + MyColor.RED;`

Correct Answer: B

QUESTION 12
Given:

```
01. public class Mud {
02.     //insert code here
03.         System.out.println("hi");
04.     }
05. }
```

And the following five fragments:

```
public static void main(String...a) {
public static void main(String.* a) {
public static void main(String... a) {
public static void main(String[]... a) {
public static void main(String...[] a) {
```

How many of the code fragments, inserted independently at line 2, compile?

A. 0
B. 1
C. 2
D. 3
E. 4
F. 5

Correct Answer: D

QUESTION 13
Given:

```
class Atom {
    Atom() { System.out.print("atom "); }
}

class Rock extends Atom {
    Rock(String type) { System.out.print(type); }
}

public class Mountain extends Rock {
    Mountain() {
        super("granite ");
        new Rock("granite ");
    }
    public static void main(String[] a) { new Mountain(); }
}
```

What is the result?

A. Compilation fails.
B. atom granite
C. granite granite
D. atom granite granite
E. An exception is thrown at runtime.
F. atom granite atom granite

Correct Answer: F

QUESTION 14
Given:

```
01. interface TestA { String toString(); }
02.
03. public class Test {
04.    public static void main(String[] args) {
05.        System.out.println(new TestA() {
06.            public String toString() { return "test"; }
07.        });
08.    }
09. }
```

What is the result?

A. test
B. null
C. An exception is thrown at runtime.
D. Compilation fails because of an error in line 1.
E. Compilation fails because of an error in line 5.
F. Compilation fails because of an error in line 6.

Correct Answer: A

QUESTION 15
Given:

```
public static void parse(String str) {
    try {
        float f = Float.parseFloat(str);
    } catch (NumberFormatException nfe) {
        f = 0;
    } finally {
        System.out.println(f);
    }
}

public static void main(String[] args) {
    parse("invalid");
}
```

What is the result?

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

Correct Answer: B

QUESTION 16
Given:

```
01. public class Blip {
02.    protected int blipvert(int x) { return 0; }
03. }
04. class Vert extends Blip {
05.    // insert code here
06. }
```

Which five methods, inserted independently at line 5, will compile? (Choose five.)

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

Correct Answer: ACEFG

QUESTION 17
Given:

```
01. class Super {
02.    private int a;
03.    protected Super(int a) { this.a = a; }
04. }

11. class Sub extends Super {
12.    public Sub(int a) { super(a); }
13.    public Sub() { this.a = 5; }
14. }
```

Which two, independently, will allow Sub to compile? (Choose two.)

A. Change line 2 to:
```
public int a;
```
B. Change line 2 to:
```
protected int a;
```
C. Change line 13 to:
```
public Sub() { this(5); }
```
D. Change line 13 to:
```
public Sub() { super(5); }
```
E. Change line 13 to:
```
public Sub() { super(a); }
```

Correct Answer: CD

QUESTION 18
Which Man class properly represents the relationship "Man has a best friend who is a Dog"?

A. `class Man extends Dog { }`
B. `class Man implements Dog { }`
C. `class Man { private BestFriend dog; }`
D. `class Man { private Dog bestFriend; }`
E. `class Man { private Dog<bestFriend>; }`
F. `class Man { private BestFriend<dog>; }`

Correct Answer: D

QUESTION 19
Given:

```
package test;

class Target {
    public String name = "hello";
}
```

What can directly access and change the value of the variable name?

A. any class
B. only the Target class
C. any class in the test package
D. any class that extends Target

Correct Answer: C

QUESTION 20
Given:

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

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

What is the result?

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

Correct Answer: D

QUESTION 21
Given:

```
05. class Building { }
06. public class Barn extends Building {
07.    public static void main(String[] args) {
08.       Building build1 = new Building();
09.       Barn barn1 = new Barn();
10.       Barn barn2 = (Barn) build1;
11.       Object obj1 = (Object) build1;
12.       String str1 = (String) build1;
13.       Building build2 = (Building) barn1;
14.   }
15. }
```

Which is true?

A. If line 10 is removed, the compilation succeeds.
B. If line 11 is removed, the compilation succeeds.
C. If line 12 is removed, the compilation succeeds.
D. If line 13 is removed, the compilation succeeds.
E. More than one line must be removed for compilation to succeed.

Correct Answer: C

QUESTION 22
A team of programmers is reviewing a proposed API for a new utility class. After some discussion, they realize that they can reduce the number of methods in the API without losing any functionality. If they implement the new design, which two OO principles will they be promoting?

A. Looser coupling
B. Tighter coupling
C. Lower cohesion
D. Higher cohesion
E. Weaker encapsulation
F. Stronger encapsulation

Correct Answer: A

QUESTION 23
Given:

```
21. class Money {
22.    private String country = "Canada";
23.    public String getC() { return country; }
24. }
25. class Yen extends Money {
26.    public String getC() { return super.country; }
27. }
28. public class Euro extends Money {
29.    public String getC(int x) { return super.getC(); }
30.    public static void main(String[] args) {
31.        System.out.print(new Yen().getC() + " " + new Euro().getC());
32.    }
33. }
```

What is the result?

A. Canada
B. null Canada
C. Canada null
D. Canada Canada
E. Compilation fails due to an error on line 26.
F. Compilation fails due to an error on line 29.

Correct Answer: E

QUESTION 24
Assuming that the serializeBanana() and the deserializeBanana() methods will correctly use Java serialization and given:

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

What is the result?

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

Correct Answer: C

QUESTION 25
Given a valid DateFormat object named df, and

```
16. Date d = new Date(0L);
17. String ds = "December 15, 2004";
18. //insert code here
```

What updates d's value with the date represented by ds?

A.
```
d = df.parse(ds);
```
B.
```
d = df.getDate(ds);
```
C.
```
try {
      d = df.parse(ds);
  } catch(ParseException e) { };
```
D.
```
try {
      d = df.getDate(ds);
  } catch(ParseException e) { };
```

Correct Answer: C

QUESTION 26
Given:

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

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

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

Correct Answer: B

QUESTION 27
Given:

```
public class TestString1 {
    public static void main(String[] args) {
        String str = "420";
        str += 42;
        System.out.print(str);
    }
}
```

What is the output?

A. 42
B. 420
C. 462
D. 42042
E. Compilation fails.
F. An exception is thrown at runtime.

Correct Answer: D

QUESTION 28
Which capability exists only in java.io.FileWriter?

A. Closing an open stream.
B. Flushing an open stream.
C. Writing to an open stream.
D. Writing a line separator to an open stream.

Correct Answer: D

QUESTION 29
Given that the current directory is empty, and that the user has read and write permissions, and the following:

```
import java.io.*;
public class DOS {
    public static void main(String[] args) {
        File dir = new File("dir");
        dir.mkdir();
        File f1 = new File(dir, "f1.txt");
        try {
            f1.createNewFile();
        } catch (IOException e) { ; }
        File newDir = new File("newDir");
        dir.renameTo(newDir);
    }
}
```

Which statement is true?

A. Compilation fails.
B. The file system has a new empty directory named dir.
C. The file system has a new empty directory named newDir.
D. The file system has a directory named dir, containing a file f1.txt.
E. The file system has a directory named newDir, containing a file f1.txt.

Correct Answer: E

QUESTION 30
Given:

```
static void test() throws RuntimeException {
    try {
        System.out.print("test ");
        throw new RuntimeException();
    }
    catch (Exception ex) { System.out.print("exception "); }
}

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

What is the result?

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

Correct Answer: D

QUESTION 31
Given:

```
public class Score implements Comparable<Score> {
    private int wins, losses;
    public Score(int w, int l) { wins = w; losses = l; }
```

```
    public int getWins() { return wins; }
    public int getLosses() { return losses; }
    public String toString() {
        return "<" + wins + "," + losses + ">";
    }
    // insert code here
}
```

Which method will complete this class?

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

Correct Answer: B

QUESTION 32
A company has a business application that provides its users with many different reports: receivables reports, payables reports, revenue projects, and so on. The company has just purchased some new, state-of-the-art, wireless printers, and a programmer has been assigned the task of enhancing all of the reports to use not only the company's old printers, but the new wireless printers as well. When the programmer starts looking into the application, the programmer discovers that because of the design of the application, it is necessary to make changes to each report to support the new printers.Which two design concepts most likely explain this situation? (Choose two.)

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

Correct Answer: BC

QUESTION 33
Given:

```
10. public class SuperCalc {
11.    protected static int multiply(int a, int b) { return a * b;}
12. }
```

and:

```
20.    public class SubCalc extends SuperCalc{
21.    public static int multiply(int a, int b) {
22.        int c = super.multiply(a, b);
23.        return c;
24.    }
25. }
```

and:

```
30.    SubCalc sc = new SubCalc ();
31. System.out.println(sc.multiply(3,4));
32. System.out.println(SubCalc.multiply(2,2));
```

What is the result?

A. 12
B. The code runs with no output.
C. An exception is thrown at runtime.

D. Compilation fails because of an error in line 21.
E. Compilation fails because of an error in line 22.
F. Compilation fails because of an error in line 31.

Correct Answer: E

QUESTION 34
Given:

```
class Foo {
    public int a = 3;
    public void addFive() { a += 5; System.out.print("f "); }
}

class Bar extends Foo {
    public int a = 8;
    public void addFive() { this.a += 5; System.out.print("b " ); }
}
```

Invoked with:

```
Foo f = new Bar();
f.addFive();
System.out.println(f.a);
```

What is the result?

A. b 3
B. b 8
C. b 13
D. f 3
E. f 8
F. f 13
G. Compilation fails.
H. An exception is thrown at runtime.

Correct Answer: A

QUESTION 35
Given:

```
22. StringBuilder sb1 = new StringBuilder("123");
23. String s1 = "123";
24. // insert code here
25. System.out.println(sb1 + " " + s1);
```

Which code fragment, inserted at line 24, outputs "123abc 123abc"?

A. `sb1.append("abc"); s1.append("abc");`
B. `sb1.append("abc"); s1.concat("abc");`
C. `sb1.concat("abc"); s1.append("abc");`
D. `sb1.concat("abc"); s1.concat("abc");`
E. `sb1.append("abc"); s1 = s1.concat("abc");`
F. `sb1.concat("abc"); s1 = s1.concat("abc");`
G. `sb1.append("abc"); s1 = s1 + s1.concat("abc");`
H. `sb1.concat("abc"); s1 = s1 + s1.concat("abc");`

Correct Answer: E

QUESTION 36

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

```
01. import java.io.*;
02. public class Foo implements Serializable {
03.     public int x, y;
04.     public Foo(int x, int y){
05.         this.x = x; this.y = y;
06.     }
07.
08.     private void writeObject(ObjectOutputStream s)
09.         throws IOException{
10.         s.writeInt(x); s.writeInt(y);
11.     }
12.
13.     private void readObject(ObjectInputStream s)
14.         throws IOException, ClassNotFoundException {
15.         //insert code here
16.     }
17. }
```

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

Correct Answer: D

QUESTION 37
Given:

```
01. public class LineUp {
02.   public static void main(String[] args) {
03.         double d = 12.345;
04.         // insert code here
05.     }
06. }
```

Which code fragment, inserted at line 4, produces the output | 12.345|?

A. `System.out.printf("|%7d| \n", d);`
B. `System.out.printf("|%7f| \n", d);`
C. `System.out.printf("|%3.7d| \n", d);`
D. `System.out.printf("|%3.7f| \n", d);`
E. `System.out.printf("|%7.3d| \n", d);`
F. `System.out.printf("|%7.3f| \n", d);`

Correct Answer: F

QUESTION 38
Given:

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

```
23. }
```

What is the result?

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

Correct Answer: D

QUESTION 39
Given:

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

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

A. `Alpha a = x;`
B. `Foo f = (Delta)x;`
C. `Foo f = (Alpha)x;`
D. `Beta b = (Beta)(Alpha)x;`

Correct Answer: B

QUESTION 40
Given:

```
public void go() {
    String o = "";
    z:
    for(int x = 0; x < 3; x++) {
        for(int y = 0; y < 2; y++) {
            if(x==1) break;
            if(x==2 && y==1) break z;
            o = o + x + y;
        }
    }
    System.out.println(o);
}
```

What is the result when the go() method is invoked?

A. 00
B. 0001
C. 000120
D. 00012021
E. Compilation fails.
F. An exception is thrown at runtime.

Correct Answer: C

QUESTION 41
Given:

```
33. try {
34.    //some code here
35. } catch (NullPointerException e1) {
36.    System.out.print("a");
37. } catch (Exception e2) {
38.    System.out.print("b");
39. } finally {
40.    System.out.print("c");
41. }
```

If some sort of exception is thrown at line 34, which output is possible?

A. a
B. b
C. c
D. ac
E. abc

Correct Answer: D

QUESTION 42
Given:

```
31. //some code here line 31
32. try {
33.     //some code here line 33
34. } catch (NullPointerException e1) {
35.     //some code here line 35
36. } finally {
37.     //some code here line 37
38. }
```

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

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

Correct Answer: BCE

QUESTION 43
Given:

```
int x = 0;
int y = 10;
do {
    y--;
    ++x;
} while (x < 5);
System.out.print(x + "," + y);
```

What is the result?

A. 5,6
B. 5,5

C. 6,5
D. 6,6

Correct Answer: B

QUESTION 44
Given:

```
public class Donkey2 {
    public static void main(String[] args) {
        boolean assertsOn = true;
        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: C

QUESTION 45
Given:

```
01. public class A{
02.   public void method1() {
03.         try {
04.             B b = new B();
05.             b.method2();
06.             //more code here
07.         } catch (TestException te){
08.             throw new RuntimeException(te);
09.         }
10.     }
11. }
```

```
01. public class B{
02.     public void method2() throws TestException {
03.         //more code here
04     }
05. }
```

```
01. class TestException extends Exception {
02. }
```

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

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

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

Correct Answer: B

QUESTION 46
Given:

```
Float pi = new Float(3.14f);
if (pi > 3) {
    System.out.print("pi is bigger than 3. ");
}
else {
    System.out.print("pi is not bigger than 3. ");
}
finally {
    System.out.println("Have a nice day.");
}
```

What is the result?

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

Correct Answer: A

QUESTION 47
Given:

```
01. public class Boxer1{
02.     Integer i;
03.     int x;
04.     public Boxer1(int y) {
05.         x = i+y;
06.         System.out.println(x);
07.     }
08.     public static void main(String[] args) {
09.         new Boxer1(new Integer(4));
10.     }
11. }
```

What is the result?

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

Correct Answer: D

QUESTION 48
Given:

```
01. public class Person {
```

```
02.     private String name;
03.     public Person(String name) { this.name = name; }
04.     public boolean equals(Person p) {
05.         return p.name.equals(this.name);
06.     }
07. }
```

Which statement is true?

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

Correct Answer: A

QUESTION 49
Which two statements are true about the hashCode method? (Choose two.)

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

Correct Answer: CE

QUESTION 50
Given a pre-generics implementation of a method:

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

What three changes allow the class to be used with generics and avoid an unchecked warning? (Choose three.)

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

Correct Answer: ACF

QUESTION 51
Given:

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

What is the result?

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

Correct Answer: C

QUESTION 52
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 53
A UNIX user named Bob wants to replace his chess program with a new one, but he is not sure where the old one is installed. Bob is currently able to run a Java chess program starting from his home directory `/home/bob` using the command:

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

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

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

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

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

Correct Answer: C

QUESTION 54
Given the following directory structure:

```
bigProject
   |--source
   |      |--Utils.java
   |
   |--classes
```

And the following command line invocation:

```
javac -d classes source/Utils.java
```

Assume the current directory is bigProject, what is the result?

A. If the compile is successful, Utils.class is added to the source directory.
B. The compiler returns an invalid flag error.
C. If the compile is successful, Utils.class is added to the classes directory.
D. If the compile is successful, Utils.class is added to the bigProject directory.

Correct Answer: C

QUESTION 55
Given:

```
package com.company.application;

public class MainClass {
    public static void main(String[] args) {}
}
```

and MainClass exists in the `/apps/com/company/application` directory. Assume the CLASSPATH environment variable is set to "." (current directory). Which two java commands entered at the command line will run MainClass? (Choose two.)

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

Correct Answer: BC

QUESTION 56

Which statement is true?

A. A class's finalize() method CANNOT be invoked explicitly.
B. super.finalize() is called implicitly by any overriding finalize() method.
C. The finalize() method for a given object is called no more than once by the garbage collector.
D. The order in which finalize() is called on two objects is based on the order in which the two objects became finalizable.

Correct Answer: C

QUESTION 57
Given:

```
public class Batman {
    int squares = 81;
    public static void main(String[] args) {
        new Batman().go();
    }
    void go() {
        incr(++squares);
        System.out.println(squares);
    }
    void incr(int squares) { squares += 10; }
}
```

What is the result?

A. 81
B. 82
C. 91
D. 92
E. Compilation fails.
F. An exception is thrown at runtime.

Correct Answer: B

QUESTION 58
Given:

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

and two separate command line invocations:

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

What is the result?

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

1 2 3

E. An exception is thrown at runtime.
2 3 4

F. An exception is thrown at runtime.
1 2 3 4

Correct Answer: B

QUESTION 59
Given:

```
public class Pass {
    public static void main(String [] args) {
        int x = 5;
        Pass p = new Pass();
        p.doStuff(x);
        System.out.print(" main x = " + x);
    }

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

What is the result?

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

Correct Answer: D

QUESTION 60
Given that c is a reference to a valid java.io.Console object, which two code fragments read a line of text from the console? (Choose two.)

A. `String s = c.readLine();`
B. `char[] c = c.readLine();`
C. `String s = c.readConsole();`
D. `char[] c = c.readConsole();`
E. `String s = c.readLine("%s", "name ");`
F. `char[] c = c.readLine("%s", "name ");`

Correct Answer: AE