Testinside

Exam : SUN 310-065

Title : Sun Certified Programmer for the Java 2 Platform. SE 6.0

Version : V5.33

Testinside - help you pass any IT exam!

Important Note, Please Read Carefully

Other TestInside products

All TestInside IT Exam Products

Our products of Offline Testing Engine

Use the offline Testing engine product to practice the questions in an exam environment.

Build a foundation of knowledge which will be useful also after passing the exam.

TestInside Testing Engine

Latest Version

We are constantly reviewing our products. New material is added and old material is revised. Free updates are available for 90 days after the purchase. You should check your member zone at TestInside and update 3-4 days before the scheduled exam date.

Here is the procedure to get the latest version:

1.Go to http://www.TestInside.com

2. Log in the **User Center**

3.The latest versions of all purchased products are downloadable from here. Just click the links.

Feedback

If you spot a possible improvement then please let us know. We always interested in improving product quality.

Feedback should be send to sales(at)TestInside.com. You should include the following: Exam number, version, page number, question number, and your login Account.

Our experts will answer your mail promptly.

Explanations

This product does not include explanations at the moment. If you are interested in providing explanations for this exam, please contact sales(at)TestInside.com.

1. Given:

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

What is the result?

A. 1 2 3

B. Compilation fails because of an error in line 12.

C. Compilation fails because of an error in line 13.

D. Compilation fails because of an error in line 14.

E. A ClassCastException is thrown at runtime.

Answer: A

2. Given:

```
11. public class Test {
12.     public enum Dogs {collie, harrier, shepherd};
13.     public static void main(String [] args) {
14.         Dogs myDog = Dogs.shepherd;
15.         switch (myDog) {
16.             case collie:
17.                 System.out.print("collie ");
18.             case default:
19.                 System.out.print("retriever ");
20.             case harrier:
21.                 System.out.print("harrier ");
22.         }
23.     }
24. }
```

What is the result?

A. harrier

B. shepherd

C. retriever

D. Compilation fails.

E. retriever harrier

F. An exception is thrown at runtime.

Answer: D

3. Given:

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

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

A. The instance gets garbage collected.

B. The code on line 33 throws an exception.

C. The code on line 35 throws an exception.

D. The code on line 31 throws an exception.

E. The code on line 33 executes successfully.

Answer: BCE

4. Given:

```
1. public class Boxer1{
2.     Integer i;
3.     int x;
4.     public Boxer1(int y) {
```

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

What is the result?

A. The value "4" is printed at the command line.

B. Compilation fails because of an error in line 5.

C. Compilation fails because of an error in line 9.

D. A NullPointerException occurs at runtime.

E. A NumberFormatException occurs at runtime.

F. An IllegalStateException occurs at runtime.

Answer: D

5. Given a valid DateFormat object named df, and

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

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

A. 18. d = df.parse(ds);

B. 18. d = df.getDate(ds);

C.
```
18. try {
19.     d = df.parse(ds);
20. }   catch(ParseException e) {   };
```

D.
```
18. try {
19.     d = df.getDate(ds);
20. }   catch(ParseException e) {   };
```

Answer: C

6. Click Next or More to continue.

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

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

Code Elements

t.start(); | t.join(); | t.pause(10); | run();
t.run(); | t.doIt(); | doIt();

Done

Answer:

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

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

Code Elements

t.start(); | t.join(); | t.pause(10); | run();
t.run(); | t.doIt(); | doIt();

Done

7. Given:

11. class ClassA {}

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

and:

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

Which three are valid? (Choose three.)

A. p0 = p1;

B. p1 = p2;

C. p2 = p4;

D. p2 = (ClassC)p1;

E. p1 = (ClassB)p3;

F. p2 = (ClassC)p4;

Answer: AEF

8. Given:

```
10: public class Hello {
11:     String title;
12:     int value;
13:     public Hello() {
14:         title += " World";
15:     }
16:     public Hello(int value) {
17:         this.value = value;
18:         title = "Hello";
19:         Hello();
20:     }
21: }
```

and:

30: Hello c = new Hello(5);

31: System.out.println(c.title);

What is the result?

A. Hello

B. Hello World

C. Compilation fails.

D. Hello World 5

E. The code runs with no output.

F. An exception is thrown at runtime.

Answer: C

9. Click Next or More to continue.

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

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

Done

Answer:

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

Function	Collection Type
java.util.Arrays	java.util.SortedSet
provides collection manipulation utilities	java.util.Arrays
defines base methods for all array objects	java.util.Iterator
java.util.Collection	java.util.TreeSet
java.util.TreeSet	java.util.Collection
java.util.SortedSet	
java.util.Iterator	
defines methods for random access to a	

Done

10. Click Next or More to continue.

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

```
import java.util.*;

public class GenericB< Place > {

   public Place foo;

   public void setFoo( Place foo) {
     this.foo = foo;
   }

   public Place getFoo() {

      return foo;
   }

  public static void main (String[] args) {
    GenericB<Cat> bar = new GenericB<Cat>();
    bar.setFoo(new Cat());
    Cat c = bar.getFoo();
  }
}

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

Code

- ? extends Pet
- T extends Pet
- ? implements Pet
- T implements Pet
- Pet extends T
- ?
- T
- <?>
- Pet

Done

Answer:

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

```
import java.util.*;

public class GenericB< T extends Pet > {

   public T foo;

   public void setFoo(T foo) {
     this.foo = foo;
   }

   public T getFoo() {

      return foo;
   }

  public static void main (String[] args) {
    GenericB<Cat> bar = new GenericB<Cat>();
    bar.setFoo(new Cat());
    Cat c = bar.getFoo();
  }
}

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

Code

- ? extends Pet
- T extends Pet
- ? implements Pet
- T implements Pet
- Pet extends T

- ?
- T
- <?>
- Pet

Done

11. Given:

11. public class Commander {

12. public static void main(String[] args) {

13. String myProp = /* insert code here */

14. System.out.println(myProp);

15. }

16. }

and the command line:

java -Dprop.custom=gobstopper Commander

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

A. System.load("prop.custom");

B. System.getenv("prop.custom");

C. System.property("prop.custom");

D. System.getProperty("prop.custom");

E. System.getProperties().getProperty("prop.custom");

Answer: DE

12. Given:

```
10. class Nav{
11.     public enum Direction { NORTH, SOUTH, EAST, WEST }
12. }
13. public class Sprite{
14.     // insert code here
15. }
```

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

A. Direction d = NORTH;

B. Nav.Direction d = NORTH;

C. Direction d = Direction.NORTH;

D. Nav.Direction d = Nav.Direction.NORTH;

Answer: D

13. Given:

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

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

A. Foo { public int bar() { return 1; } }

B. new Foo { public int bar() { return 1; } }

C. new Foo() { public int bar() { return 1; } }

D. new class Foo { public int bar() { return 1; } }

Answer: C

14. 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[] argv ) {
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.

Answer: BEF

15. Given:

```
1. class TestA {
2.     public void start() { System.out.println("TestA"); }
3. }
4. public class TestB extends TestA {
5.     public void start() { System.out.println("TestB"); }
6.     public static void main(String[] args) {
7.         ((TestA)new TestB()).start();
8.     }
9. }
```

What is the result?

A. TestA

B. TestB

C. Compilation fails.

D. An exception is thrown at runtime.

Answer: B

16. Given:

```
1. interface TestA { String toString(); }
2. public class Test {
3.     public static void main(String[] args) {
4.         System.out.println(new TestA() {
5.             public String toString() { return "test"; }
6.         });
```

```
7.     }
8. }
```

What is the result?

A. test

B. null

C. An exception is thrown at runtime.

D. Compilation fails because of an error in line 1.

E. Compilation fails because of an error in line 4.

F. Compilation fails because of an error in line 5.

Answer: A

17. Given:

```
11. public interface A111 {
12.     String s = "yo";
13.     public void method1();
14. }
17. interface B { }
20. interface C extends A111, B {
21.     public void method1();
22.     public void method1(int x);
23. }
```

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.

Answer: A

18. Given:

```
11. public interface A { public void m1(); }
12.
13. class B implements A { }
14. class C implements A { public void m1() { } }
15. class D implements A { public void m1(int x) { } }
16. abstract class E implements A { }
17. abstract class F implements A { public void m1() { } }
18. abstract class G implements A { public void m1(int x) { } }
```

What is the result?

A. Compilation succeeds.

B. Exactly one class does NOT compile.

C. Exactly two classes do NOT compile.

D. Exactly four classes do NOT compile.

E. Exactly three classes do NOT compile.

Answer: C

19. Given:

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

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

A. double getSalesAmount() { return 1230.45; }

B. public double getSalesAmount() { return 1230.45; }

C. private double getSalesAmount() { return 1230.45; }

D. protected double getSalesAmount() { return 1230.45; }

Answer: BD

20. Given:

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

Which class correctly uses the Data interface and Info class?

A.
```
public class Employee extends Info implements Data {
    public void load() { /*do something*/ }
}
```

B.
```
public class Employee implements Info extends Data {
    public void load() { /*do something*/ }
}
```

C.
```
public class Employee extends Info implements Data {
   public void load(){ /*do something*/ }
   public void Info.load(){ /*do something*/ }
}
```

D.
```
public class Employee implements Info extends Data {
   public void Data.load(){ /*do something*/ }
   public void load(){ /*do something*/ }
}
```

E.
```
public class Employee implements Info extends Data {
   public void load(){ /*do something*/ }
   public void Info.load(){ /*do something*/ }
}
```

F.
```
public class Employee extends Info implements Data{
   public void Data.load() { /*do something*/ }
   public void Info.load() { /*do something*/ }
}
```

Answer: A

21. Click Next or More to continue.

Place the code fragments in position to complete the Displayable interface.

```
interface Reloadable {
  public void reload();
}

class Edit {
  public void edit() { /* Edit Here */ }
}

interface Displayable
```

Place here | Place here | {

Place here

```
}
```

Code Fragments

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

Done

Answer:

Place the code fragments in position to complete the Displayable interface.

```
interface Reloadable {
  public void reload();
}

class Edit {
  public void edit() { /* Edit Here */ }
}

interface Displayable
```

extends | Reloadable | {

public void display();

```
}
```

Code Fragments

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

Done

22. Given:

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

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

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

Answer: BE

23. Given

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

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

A. final

B. static

C. native

D. public

E. private

F. abstract

G. protected

Answer: ABD

24. 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() { System.out.println("go in Sente."); }
11. }
```

```
12.
13. class Goban extends Sente {
14.     public void go() { System.out.println("go in Goban"); }
15. }
16.
17. class Stone extends Goban implements Go { }
18.
19. 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.

Answer: C

25. Given:

```
11. public static void parse(String str) {
12.     try {
13.         float f = Float.parseFloat(str);
14.     } catch (NumberFormatException nfe) {
15.         f = 0;
```

```
16.     } finally {
17.         System.out.println(f);
18.     }
19. }
20. public static void main(String[] args) {
21.     parse("invalid");
22. }
```

What is the result?

A. 0.0

B. Compilation fails.

C. A ParseException is thrown by the parse method at runtime.

D. A NumberFormatException is thrown by the parse method at runtime.

Answer: B

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

A. static final int[] a = { 100,200 };

B. static final int[] a;

static { a=new int[2]; a[0]=100; a[1]=200; }

C. static final int[] a = new int[2]{ 100,200 };

D. static final int[] a;

static void init() { a = new int[3]; a[0]=100; a[1]=200; }

Answer: AB

27. Given:

```
11. public enum Title {
12.     MR("Mr."), MRS("Mrs."), MS("Ms.");
13.     private final String title;
14.     private Title(String t) { title = t; }
15.     public String format(String last, String first) {
16.         return title + " " + first + " " + last;
```

```
17.     }
18. }
19. public static void main(String[] args) {
20.     System.out.println(Title.MR.format("Doe", "John"));
21. }
```

What is the result?

A. Mr. John Doe

B. An exception is thrown at runtime.

C. Compilation fails because of an error in line 12.

D. Compilation fails because of an error in line 15.

E. Compilation fails because of an error in line 20.

Answer: A

28. Click Next or More to continue.

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

```
enum Element {
```

1st

2nd

3rd

4th

5th

Lines

```
public String info() { return "element"; }
};
FIRE { public String info() { return "Hot"; }
EARTH, WIND,
}
```

Done

Answer:

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

```
enum Element {
EARTH, WIND,
FIRE { public String info() { return "Hot"; }
};
public String info() { return "element"; }
}
```

Lines

```
public String info() { return "element"; }
};
FIRE { public String info() { return "Hot"; }
EARTH, WIND,
}
```

Done

29. Given:

```
1. class Alligator {
2.     public static void main(String[] args) {
3.         int []x[] = {{1,2}, {3,4,5}, {6,7,8,9}};
4.         int [][]y = x;
5.         System.out.println(y[2][1]);
6.     }
7. }
```

What is the result?

A. 2

B. 3

C. 4

D. 6

E. 7

F. Compilation fails.

Answer: E

30. Given:

```
11. public class Rainbow {
12.     public enum MyColor {
13.         RED(0xff0000), GREEN(0x00ff00), BLUE(0x0000ff);
14.         private final int rgb;
15.         MyColor(int rgb) { this.rgb = rgb; }
16.         public int getRGB() { return rgb; }
17.     };
18.     public static void main(String[] args) {
19.         // insert code here
20.     }
21. }
```

Which code fragment, inserted at line 19, 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;

Answer: B

31. Given:

```
10. class One {
11.     public One foo() { return this; }
12. }
13. class Two extends One {
14.     public One foo() { return this; }
15. }
```

```
16. class Three extends Two {
17.     // insert method here
18. }
```

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

A. public void foo() {}

B. public int foo() { return 3; }

C. public Two foo() { return this; }

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

E. public Object foo() { return this; }

Answer: CD

32. Given:

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

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

A. int foo() { /* more code here */ }

B. void foo() { /* more code here */ }

C. public void foo() { /* more code here */ }

D. private void foo() { /* more code here */ }

E. protected void foo() { /* more code here */ }

Answer: BCE

33. Click the Exhibit button.

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

```
1. public class AImpl implements A {
2.     public void doSomething(String msg) { }
3. }
1. public class B {
2.     public A doit() {
3.         // more code here
4.     }
5.
6.     public String execute() {
7.         // more code here
8.     }
9. }
1. public class C extends B {
2.     public AImpl doit() {
3.         // more code here
4.     }
5.
6.     public Object execute() {
7.         // more code here
8.     }
9. }
```

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

A. Compilation will succeed for all classes and interfaces.

B. Compilation of class C will fail because of an error in line 2.

C. Compilation of class C will fail because of an error in line 6.

D. Compilation of class AImpl will fail because of an error in line 2.

Answer: C

34. Given:

```
1. public class GC {
```

```
2.    private Object o;
3.    private void doSomethingElse(Object obj) { o = obj; }
4.    public void doSomething() {
5.        Object o = new Object();
6.        doSomethingElse(o);
7.        o = new Object();
8.        doSomethingElse(null);
9.        o = null;
10.   }
11. }
```

When the doSomething method is called, after which line does the Object created in line 5 become available for garbage collection?

A. Line 5

B. Line 6

C. Line 7

D. Line 8

E. Line 9

F. Line 10

Answer: D

35. 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.

Answer: C

36. Click the Exhibit button.

Given the fully-qualified class names:

com.foo.bar.Dog

com.foo.bar.blatz.Book

com.bar.Car

com.bar.blatz.Sun

Which graph represents the correct directory structure for a JAR file from which those classes can be used by the compiler and JVM?

A. Jar A

B. Jar B

C. Jar C

D. Jar D

E. Jar E

Answer: A

37. A developer is creating a class Book, that needs to access class Paper. The Paper class is deployed in a JAR named myLib.jar.

Which three, taken independently, will allow the developer to use the Paper class while compiling the Book class? (Choose three.)

A. The JAR file is located at $JAVA_HOME/jre/classes/myLib.jar.

B. The JAR file is located at $JAVA_HOME/jre/lib/ext/myLib.jar..

C. The JAR file is located at /foo/myLib.jar and a classpath environment variable is set that includes /foo/myLib.jar/Paper.class.

D. The JAR file is located at /foo/myLib.jar and a classpath environment variable is set that includes /foo/myLib.jar.

E. The JAR file is located at /foo/myLib.jar and the Book class is compiled using javac -cp /foo/myLib.jar/Paper Book.java.

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

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

Answer: BDG

38. Given:

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

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

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

Answer: BC

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

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

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

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

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

A. /test/Chess.class

B. /home/bob/Chess.class

C. /test/games/Chess.class

D. /usr/lib/games/Chess.class

E. /home/bob/games/Chess.class

F. inside jarfile /opt/java/lib/Games.jar (with a correct manifest)

G. inside jarfile /home/bob/downloads/Games.jar (with a correct manifest)

Answer: C

40. Given:

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

and the invocation:

```
21. test("four");
```

```
22. test("tee");
23. test("to");
```

What is the result?

A. r, t, t,

B. r, e, o,

C. Compilation fails.

D. An exception is thrown at runtime.

Answer: C

41. Given:

```
3. public class Spock {
4.     public static void main(String[] args) {
5.         Long tail = 2000L;
6.         Long distance = 1999L;
7.         Long story = 1000L;
8.         if((tail > distance)    ((story * 2) == tail))
9.             System.out.print("1");
10.         if((distance + 1 != tail)    ((story * 2) == distance))
11.             System.out.print("2");
12.     }
13. }
```

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.

Answer: E

42. Given:

```
3. interface Fish { }
4. class Perch implements Fish { }
5. class Walleye extends Perch { }
6. class Bluegill { }
7. public class Fisherman {
8.     public static void main(String[] args) {
9.         Fish f = new Walleye();
10.         Walleye w = new Walleye();
11.         Bluegill b = new Bluegill();
12.         if(f instanceof Perch)          System.out.print("f-p ");
13.         if(w instanceof Fish)           System.out.print("w-f ");
14.         if(b instanceof Fish)           System.out.print("b-f ");
15.     }
16. }
```

What is the result?

A. w-f

B. f-p w-f

C. w-f b-f

D. f-p w-f b-f

E. Compilation fails.

F. An exception is thrown at runtime.

Answer: B

43. Click the Exhibit button.

```
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. }
```

Given:

```
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.

Answer: A

44. Given:

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

What is the result?

A. An exception is thrown at runtime.

B. Compilation fails because of an error in line 7.

C. Compilation fails because of an error in line 4.

D. Compilation succeeds and no runtime errors with class A occur.

Answer: C

45. Click Next or More to continue.

Add methods to the Beta class to make it compile correctly.

```
class Alpha {
  public void bar( int... x ) { }
  public void bar( int x ) { }
}

public class Beta extends Alpha {

  Place here

  Place here

  Place here

}
```

Done

Methods

- private void bar(int x) { }
- public void bar(int x) { }
- public int bar(String x) { return 1; }
- public Alpha bar(int x) { }
- public void bar(int x, int y) { }
- public int bar(int x) { return x; }

Answer:

Add methods to the Beta class to make it compile correctly.

```
class Alpha {
  public void bar( int... x ) { }
  public void bar( int x ) { }
}

public class Beta extends Alpha {

  public void bar( int x ) { }

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

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

}
```

Done

Methods

- private void bar(int x) { }
- public void bar(int x) { }
- public int bar(String x) { return 1; }
- public Alpha bar(int x) { }
- public void bar(int x, int y) { }
- public int bar(int x) { return x; }

46. 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) {
```

```
17.         Alpha a = new Beta();
18.         Beta b = (Beta)a;
19.         a.foo();
20.         b.foo();
21.     }
22. }
```

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.

Answer: D

47. Given:

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

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

A. Point p = new Point();

B. Line.Point p = new Line.Point();

C. The Point class cannot be instatiated at line 15.

D. Line l = new Line() ; l.Point p = new l.Point();

Answer: B

48. Given:

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

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

A. Point p = Line.getPoint();

B. Line.Point p = Line.getPoint();

C. Point p = (new Line()).getPoint();

D. Line.Point p = (new Line()).getPoint();

Answer: D

49. Click the Exhibit button.

```
11. class Person {
12.     String name = "No name";
13.     public Person(String nm) { name = nm; }
14. }
15.
16. class Employee extends Person {
17.     String empID = "0000";
18.     public Employee(String id) { empID = 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. }
```

What is the result?

A. 4321

B. 0000

C. An exception is thrown at runtime.

D. Compilation fails because of an error in line 18.

Answer: D

50. Given:

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

Which statement is true?

A. The code will compile without changes.

B. The code will compile if public Tree() { Plant(); } is added to the Tree class.

C. The code will compile if public Plant() { Tree(); } is added to the Plant class.

D. The code will compile if public Plant() { this("fern"); } is added to the Plant class.

E. The code will compile if public Plant() { Plant("fern"); } is added to the Plant class.

Answer: D

51. Given:

```
21. abstract class C1 {
22.     public C1() { System.out.print(1); }
23. }
24. class C2 extends C1 {
25.     public C2() { System.out.print(2); }
26. }
27. class C3 extends C2 {
28.     public C3() { System.out.println(3); }
29. }
30. public class Ctest {
31.     public static void main(String[] a) { new C3(); }
32. }
```

What is the result?

A. 3

B. 23

C. 32

D. 123

E. 321

F. Compilation fails.

G. An exception is thrown at runtime.

Answer: D

52. Given:

```
5. class Atom {
6.     Atom() { System.out.print("atom "); }
7. }
8. class Rock extends Atom {
9.     Rock(String type) { System.out.print(type); }
10. }
11. public class Mountain extends Rock {
```

```
12.     Mountain() {
13.         super("granite ");
14.         new Rock("granite ");
15.     }
16.     public static void main(String[] a) { new Mountain(); }
17. }
```

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

Answer: F

53. Given:

```
12. public class Test {
13.     public enum Dogs {collie, harrier};
14.     public static void main(String [] args) {
15.         Dogs myDog = Dogs.collie;
16.         switch (myDog) {
17.             case collie:
18.                 System.out.print("collie ");
19.             case harrier:
20.                 System.out.print("harrier ");
21.         }
22.     }
23. }
```

What is the result?

A. collie

B. harrier

C. Compilation fails.

D. collie harrier

E. An exception is thrown at runtime.

Answer: D

54. Given:

```
public class Test {
    public static void main(String [] args) {
        int x = 5;
        boolean b1 = true;
        boolean b2 = false;

        if ((x == 4) && !b2 )
            System.out.print("1 ");
            System.out.print("2 ");
        if ((b2 = true) && b1 )
            System.out.print("3 ");
    }
}
```

What is the result?

A. 2

B. 3

C. 1 2

D. 2 3

E. 1 2 3

F. Compilation fails.

G. An exception is thrown at runtime.

Answer: D

55. Given:

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

What is the result when method testIfA is invoked?

A. True

B. Not true

C. An exception is thrown at runtime.

D. Compilation fails because of an error at line 12.

E. Compilation fails because of an error at line 19.

Answer: A

56. Given:

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

What is the result?

A. three

B. other

C. An exception is thrown at runtime.

D. Compilation fails because of an error on line 12.

E. Compilation fails because of an error on line 13.

F. Compilation fails because of an error on line 15.

Answer: A

57. Given:

```
11. public static void main(String[] args) {
12.     String str = "null";
13.     if (str == null) {
14.         System.out.println("null");
15.     } else (str.length() == 0) {
16.         System.out.println("zero");
17.     } else {
18.         System.out.println("some");
19.     }
20. }
```

What is the result?

A. null

B. zero

C. some

D. Compilation fails.

E. An exception is thrown at runtime.

Answer: D

58. Given:

```
11. Float pi = new Float(3.14f);
12. if (pi > 3) {
13.     System.out.print("pi is bigger than 3. ");
14. }
```

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

What is the result?

A. Compilation fails.

B. pi is bigger than 3.

C. An exception occurs at runtime.

D. pi is bigger than 3. Have a nice day.

E. pi is not bigger than 3. Have a nice day.

Answer: A

59. Given:

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

What is the result?

A. 3, 2, 1,

B. 1, 2, 3,

C. Compilation fails.

D. The code runs with no output.

E. An exception is thrown at runtime.

Answer: C

60. 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.

Answer: B

61. Given:

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

What is the result?

A. 6

B. 7

C. 10

D. 11

E. Compilation fails.

F. An exception is thrown at runtime.

Answer: E

62. Given:

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

What is the result?

A. 2

B. 24

C. 234

D. 246

E. 2346

F. Compilation fails.

Answer: B

63. Given:

```
22.   public void go() {
23.       String o = "";
24.       z:
25.       for(int x = 0; x < 3; x++) {
26.           for(int y = 0; y < 2; y++) {
27.               if(x==1) break;
28.               if(x==2 && y==1) break z;
29.               o = o + x + y;
30.           }
31.       }
32.       System.out.println(o);
33.   }
```

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.

Answer: C

64. 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.     void go(int[]... z) {
```

```
8.      for(int[] a : z)
9.          System.out.print(a[0]);
10.     }
11. }
```

What is the result?

A. 1

B. 12

C. 14

D. 123

E. Compilation fails.

F. An exception is thrown at runtime.

Answer: C

65. Given:

```
3. public class Breaker {
4.     static String o = "";
5.     public static void main(String[] args) {
6.         z:
7.         o = o + 2;
8.         for(int x = 3; x < 8; x++) {
9.             if(x==4) break;
10.            if(x==6) break z;
11.            o = o + x;
12.        }
13.     System.out.println(o);
14.     }
15. }
```

What is the result?

A. 23

B. 234

C. 235

D. 2345

E. 2357

F. 23457

G. Compilation fails.

Answer: G

66. Given:

```
1. public class Donkey {
2.     public static void main(String[] args) {
3.         boolean assertsOn = false;
4.         assert (assertsOn) : assertsOn = true;
5.         if(assertsOn) {
6.             System.out.println("assert is on");
7.         }
8.     }
9. }
```

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

no output

B. no output

assert is on

C. assert is on

assert is on

D. no output

An AssertionError is thrown.

E. assert is on

An AssertionError is thrown.

Answer: D

67. 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

Answer: B

68. Given:

```
11.   public void go(int x) {
12.       assert (x > 0);
13.       switch(x) {
14.           case 2: ;
15.           default: assert false;
16.       }
17.   }
18.   private void go2(int x) {    assert (x < 0); }
```

Which statement is true?

A. All of the assert statements are used appropriately.

B. Only the assert statement on line 12 is used appropriately.

C. Only the assert statement on line 15 is used appropriately.

D. Only the assert statement on line 18 is used appropriately.

E. Only the assert statements on lines 12 and 15 are used appropriately.

F. Only the assert statements on lines 12 and 18 are used appropriately.

G. Only the assert statements on lines 15 and 18 are used appropriately.

Answer: G

69. Given:

```
11. class A {
12.     public void process() { System.out.print("A,"); }}
13. class B extends A {
14.     public void process() throws IOException {
15.         super.process();
16.         System.out.print("B,");
17.         throw new IOException();
18.     }
19. public static void main(String[] args) {
20.     try { new B().process(); }
21.     catch (IOException e) { System.out.println("Exception"); }}}
22. }
```

What is the result?

A. Exception

B. A,B,Exception

C. Compilation fails because of an error in line 20.

D. Compilation fails because of an error in line 14.

E. A NullPointerException is thrown at runtime.

Answer: D

70. Given:

```
11. static class A {
12.         void process() throws Exception { throw new Exception(); }
```

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

What is the result?

A. B

B. The code runs with no output.

C. Compilation fails because of an error in line 12.

D. Compilation fails because of an error in line 15.

E. Compilation fails because of an error in line 18.

Answer: A

71. Given:

```
1. class TestException extends Exception { }
2. class A {
3.     public String sayHello(String name) throws TestException {
4.         if(name == null)    throw new TestException();
5.         return "Hello " + name;
6.     }
7. }
8. public class TestA {
9.     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.

Answer: D

72. 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

Answer: D

73. 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.

Answer: A

74. Given:

```
5. class A {
6.     void foo() throws Exception { throw new Exception(); }
7. }
8. class SubB2 extends A {
9.     void foo() { System.out.println("B "); }
10. }
11. class Tester {
12.     public static void main(String[] args) {
13.         A a = new SubB2();
14.         a.foo();
15.     }
16. }
```

What is the result?

A. B

B. B, followed by an Exception.

C. Compilation fails due to an error on line 9.

D. Compilation fails due to an error on line 14.

E. An Exception is thrown with no other output.

Answer: D

75. 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.

Answer: BE

76. Click the Exhibit button.

```
1. public class A {
2.     public void method1() {
3.         try {
4.             B b = new B();
5.             b.method2();
6.             // more code here
7.         } catch (TestException te) {
8.             throw new RuntimeException(te);
9.         }
6.     }
7. }
```

```
1. public class B {
2.     public void method2() throws TestException {
3.         // more code here
4.     }
5. }
```

```
1. public class TestException extends Exception {
2. }
```

Given:

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

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

A. Line 33 must be called within a try block.

B. The exception thrown by method1 in class A is not required to be caught.

C. The method declared on line 31 must be declared to throw a RuntimeException.

D. On line 5 of class A, the call to method2 of class B does not need to be placed in a try/catch block.

Answer: B

77. Given:

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

What is the result?

A. test

B. Exception

C. Compilation fails.

D. NullPointerException

Answer: C

78. Given:

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

What is the result?

A. end

B. Compilation fails.

C. exception end

D. exception test end

E. A Throwable is thrown by main.

F. An Exception is thrown by main.

Answer: E

79. Given:

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

```
20.     catch (Exception ex) { System.out.print("exception "); }
21. }
```

What is the result?

A. null

B. finally

C. null finally

D. Compilation fails.

E. finally exception

Answer: E

80. Given:

```
11. static void test() throws RuntimeException {
12.     try {
13.         System.out.print("test ");
14.         throw new RuntimeException();
15.     }
16.     catch (Exception ex) { System.out.print("exception "); }
17. }
18. public static void main(String[] args) {
19.     try { test(); }
20.     catch (RuntimeException ex) { System.out.print("runtime "); }
21.     System.out.print("end ");
22. }
```

What is the result?

A. test end

B. Compilation fails.

C. test runtime end

D. test exception end

E. A Throwable is thrown by main at runtime.

Answer: D

81. 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 AssertionException("value is null");

}

D. if (value == null) {

   throw new IllegalArgumentException("value is null");

}

Answer: D

82. Click the Exhibit button.

```
10. public class ClassA {
11.     public void methodA() {
12.         ClassB classB = new ClassB();
13.         classB.getValue();
14.     }
15. }
```

And:

```
20. class ClassB {
21.     public ClassC classC;
22.
23.     public String getValue() {
```

```
24.         return classC.getValue();
25.     }
26. }
```

And:

```
30. class ClassC {
31.     public String value;
32.
33.     public String getValue() {
34.         value = "ClassB";
35.         return value;
36.     }
37. }
```

Given:

```
ClassA a = new ClassA();
a.methodA();
```

What is the result?

A. Compilation fails.

B. ClassC is displayed.

C. The code runs with no output.

D. An exception is thrown at runtime.

Answer: D

83. Given:

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

```
17.     }
18. }
```

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

A. Alpha a = x;

B. Foo f = (Delta)x;

C. Foo f = (Alpha)x;

D. Beta b = (Beta)(Alpha)x;

Answer: B

84. Given:

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

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

A. java.lang.StackOverflowError

B. java.lang.IllegalStateException

C. java.lang.ExceptionInInitializerError

D. java.lang.ArrayIndexOutOfBoundsException

Answer: C

85. Which two code fragments are most likely to cause a StackOverflowError? (Choose two.)

A. int []x = {1,2,3,4,5};

```
for(int y = 0; y < 6; y++)
  System.out.println(x[y]);
```

B. static int[] x = {7,6,5,4};

```
static    { x[1] = 8;
              x[4] = 3; }
```

C. for(int y = 10; y < 10; y++)

doStuff(y);

D. void doOne(int x) { doTwo(x); }

void doTwo(int y) { doThree(y); }

void doThree(int z) { doTwo(z); }

E. for(int x = 0; x < 1000000000; x++)

doStuff(x);

F. void counter(int i) { counter(++i); }

Answer: DF

86. Given:

```
1. public class TestString1 {
2.     public static void main(String[] args) {
3.         String str = "420";
4.         str += 42;
5.         System.out.print(str);
6.     }
7. }
```

What is the output?

A. 42

B. 420

C. 462

D. 42042

E. Compilation fails.

F. An exception is thrown at runtime.

Answer: D

87. 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.

Answer: D

88. Given:

```
1. public class TestString3 {
2.     public static void main(String[] args) {
3.         // insert code here

5.         System.out.println(s);
6.     }
7. }
```

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

A. String s = "123456789";

s = (s-"123").replace(1,3,"24") - "89";

B. StringBuffer s = new StringBuffer("123456789");

s.delete(0,3).replace(1,3,"24").delete(4,6);

C. StringBuffer s = new StringBuffer("123456789");

s.substring(3,6).delete(1,3).insert(1, "24");

D. StringBuilder s = new StringBuilder("123456789");

s.substring(3,6).delete(1,2).insert(1, "24");

E. StringBuilder s = new StringBuilder("123456789");

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

Answer: BE

89. 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.

Answer: B

90. 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.

Answer: AB

91. 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");

Answer: E

92. Given:

```
1. public class BuildStuff {
2.     public static void main(String[] args) {
3.         Boolean test = new Boolean(true);
4.         Integer x = 343;
5.         Integer y = new BuildStuff().go(test, x);
6.         System.out.println(y);
7.     }
8.     int go(Boolean b, int i) {
9.         if(b) return (i/7);
10.         return (i/49);
11.     }
```

12. }

What is the result?

A. 7

B. 49

C. 343

D. Compilation fails.

E. An exception is thrown at runtime.

Answer: B

93. Click Next or More to continue.

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

reader = Place here Place here "in"));

writer = Place here Place here Place here "out")));

Constructors

new FileReader(new PrintReader(new BufferedReader(

new BufferedWriter(new FileWriter(new PrintWriter(

Done

Answer:

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

reader = new BufferedReader(new FileReader("in"));

writer = new PrintWriter(new BufferedWriter(new FileWriter("out")));

Constructors

new FileReader(new PrintReader(new BufferedReader(

new BufferedWriter(new FileWriter(new PrintWriter(

Done

94. Click Next or More to continue.

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

Place the code fragments in position to complete this method.

```
public static boolean doesFileExist(String[] directories, String filename) {
    Place here
    for ( String dir : directories ) {
        Place here
    }
    Place here

    Place here
}
```

Code Fragments

path =	return ! file.isNew();	return (file != null);
String path = "";	path = path.getFile(filename);	File path = new File("");
return file.exists();	return path.isFile();	File file = new File(path,
path = new File(path, dir);	File path = new	path = path

Done

Answer:

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

Place the code fragments in position to complete this method.

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

    return file.exists();
}
```

Code Fragments

path =	return ! file.isNew();	return (file != null);
String path = "";	path = path.getFile(filename);	File path = new File("");
return file.exists();	return path.isFile();	File file = new File(path,
path = new File(path, dir);	File path = new	path = path

Done

95. Click Next or More to continue.

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

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

      while( Place here  Place here  ){
        System.out.println(temp);
      }
    } catch Place here
      e.printStackTrace();
    }
  }
}
```

Code Fragments

(temp =	& & buffReader.hasNext()
(temp =	(IOException e) {
!= null	(FileNotFoundException e)

Done

Answer:

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

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

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

Code Fragments

(temp =	& & buffReader.hasNext()
(temp =	(IOException e) {
!= null	(FileNotFoundException e)

Done

96. Click Next or More to continue.

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

Program

```
import java.io.*

public class ReadFile {
  public static void main(String [] args) {

    try {
      File  ?  = new File("MyText.txt");
      Place here   ?  = new  Place here  (x1);
      Place here   x4 = new  Place here  (x2);
      String x3 = null;
      while (( x3 =  ?  .Place here  ()) != null) {
        System.out.println(x3);
      } ?  .Place here  ();
    } catch(Exception ex) {
        ex.printStackTrace();
    }
  }
}
```

Code Fragments

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

Answer:

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

Program

```
import java.io.*

public class ReadFile {
  public static void main(String [] args) {

    try {
      File  x1  = new File("MyText.txt");
      FileReader   x2  = new  FileReader  (x1);
      BufferedReader   x4 = new  BufferedReader  (x2);
      String x3 = null;
      while (( x3 =  x4  .readLine  ()) != null) {
        System.out.println(x3);
      } x4  .close  ();
    } catch(Exception ex) {
        ex.printStackTrace();
    }
  }
}
```

Code Fragments

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

97. Given that the current directory is empty, and that the user has read and write privileges to the current directory, and the following:

1. import java.io.*;

2. public class Maker {

3. public static void main(String[] args) {

4. File dir = new File("dir");

```
5.        File f = new File(dir, "f");
6.    }
7. }
```

Which statement is true?

A. Compilation fails.

B. Nothing is added to the file system.

C. Only a new file is created on the file system.

D. Only a new directory is created on the file system.

E. Both a new file and a new directory are created on the file system.

Answer: B

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

```
11. import java.io.*;
12. public class DOS {
13.     public static void main(String[] args) {
14.         File dir = new File("dir");
15.         dir.mkdir();
16.         File f1 = new File(dir, "f1.txt");
17.         try {
18.             f1.createNewFile();
19.         } catch (IOException e) { ; }
20.         File newDir = new File("newDir");
21.         dir.renameTo(newDir);
22.     }
23. }
```

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.

Answer: E

99. Given:

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

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

A. The value of b is 2.

B. The value of a is 3.14.

C. The value of b is 2.00.

D. The value of a is 3.141.

E. The value of a is 3.1415.

F. The value of a is 3.1416.

G. The value of b is 2.0000.

Answer: CF

100. 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.

Answer: D

101. 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));

Answer: B

102. Given:

5. import java.util.Date;

6. import java.text.DateFormat;

21. DateFormat df;

22. Date date = new Date();

23. // insert code here

24. String s = df.format(date);

Which code fragment, inserted at line 23, allows the code to compile?

A. df = new DateFormat();

B. df = Date.getFormat();

C. df = date.getFormat();

D. df = DateFormat.getFormat();

E. df = DateFormat.getInstance();

Answer: E

103. Given:

11. String test = "a1b2c3";

12. String[] tokens = test.split("\\d");

13. for(String s: tokens) System.out.print(s + " ");

What is the result?

A. a b c

B. 1 2 3

C. a1b2c3

D. a1 b2 c3

E. Compilation fails.

F. The code runs with no output.

G. An exception is thrown at runtime.

Answer: A

104. Given:

11. String test = "Test A. Test B. Test C.";

12. // insert code here

13. String[] result = test.split(regex);

Which regular expression, inserted at line 12, correctly splits test into "Test A", "Test B", and "Test C"?

A. String regex = "";

B. String regex = " ";

C. String regex = ".*";

D. String regex = "\\s";

E. String regex = "\\.\\s*";

F. String regex = "\\w[\.] +";

Answer: E

105. Click Next or More to continue.

Given:

```
System.out.printf("Pi is approximately %f and E is approximately %b",
                  Math.PI. Math.E):
```

Place the values where they would appear in the output.

Pi is approximately [Place here]

and E is approximately [Place here]

Values

3	3.141593	true	Math.PI
2	2.718282	false	Math.E

Done

Answer:

Given:

```
System.out.printf("Pi is approximately %f and E is approximately %b",
                  Math.PI. Math.E):
```

Place the values where they would appear in the output.

Pi is approximately [3.141593]

and E is approximately [true]

Values

3	3.141593	true	Math.PI
2	2.718282	false	Math.E

Done

106. Given:

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

What is the result?

A. Compilation fails.

B. After line 15, the value of age is 5.

C. After line 15, the value of age is 3.

D. An exception is thrown at runtime.

Answer: D

107. Click Next or More to continue.

Place the code fragments into position to produce the output:

true true false

Code

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

while ( Place here ) {

  if ( Place here ) {

    System.out.print( Place here + " ");

  } else Place here ;
}
```

Code Fragments

scanner.hasNextBoolean | scanner.nextBoolean() | scanner.next() | scanner.hasNext() | Done

Answer:

Place the code fragments into position to produce the output:

true true false

Code

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

while ( scanner.hasNext() ) {

  if ( scanner.hasNextBoolean ) {

    System.out.print( scanner.nextBoolean() + " ");

  } else scanner.next() ;
}
```

Code Fragments

scanner.hasNextBoolean | scanner.nextBoolean() | scanner.next() | scanner.hasNext() | Done

108. Given:

```
1. public class LineUp {
2.     public static void main(String[] args) {
3.         double d = 12.345;
4.         // insert code here
5.     }
6. }
```

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);

Answer: F

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

A.
```
new Thread() {
   public void run() { doStuff(); }
};
```

B.
```
new Thread() {
   public void start() { doStuff(); }
};
```

C.
```
new Thread() {
   public void start() { doStuff(); }
}.run();
```

D.
```
new Thread() {
   public void run() { doStuff(); }
}.start();
```

E.
```
new Thread(new Runnable() {
```

```
   public void run() { doStuff(); }
}).run();
```

F. new Thread(new Runnable() {

```
   public void run() { doStuff(); }
}).start();
```

Answer: DF

110. Given:

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

What is the result?

A. Compilation fails.

B. An exception is thrown at runtime.

C. The code executes and prints "running".

D. The code executes and prints "runningrunning".

E. The code executes and prints "runningrunningrunning".

Answer: E

111. Given:

```
1. public class Threads4 {
2.     public static void main (String[] args) {
```

```
3.        new Threads4().go();
4.    }
5.    public void go() {
6.        Runnable r = new Runnable() {
7.            public void run() {
8.                System.out.print("foo");
9.            }
10.       };
11.       Thread t = new Thread(r);
12.       t.start();
13.       t.start();
14.   }
15. }
```

What is the result?

A. Compilation fails.

B. An exception is thrown at runtime.

C. The code executes normally and prints "foo".

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

Answer: B

112. Given:

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

What is the result?

A. Compilation fails.

B. An exception is thrown at runtime.

C. The code executes normally and prints "bar".

D. The code executes normally, but nothing prints.

Answer: C

113. Click the Exhibit button.

Given:

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

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

A. 4

B. 5

C. 8

D. 9

E. Compilation fails.

F. An exception is thrown at runtime.

G. It is impossible to determine for certain.

Answer: D

114. Given:

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

Which two can be results? (Choose two.)

A. java.lang.RuntimeException: Problem

B. run.

java.lang.RuntimeException: Problem

C. End of method.

java.lang.RuntimeException: Problem

D. End of method.

run.

java.lang.RuntimeException: Problem

E. run.

java.lang.RuntimeException: Problem

End of method.

Answer: DE

115. Given:

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

What is the result?

A. Compilation fails.

B. An exception is thrown at runtime.

C. The code executes normally and prints "sleep".

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

Answer: C

116. Given:

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

What can be a result?

A. Compilation fails.

B. An exception is thrown at runtime.

C. The code executes and prints "StartedComplete".

D. The code executes and prints "StartedComplete0123".

E. The code executes and prints "Started0123Complete".

Answer: E

117. Click Next or More to continue.

Give

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

Assume that sleep(n) executes in exactly n milliseconds, and all other code executes in an insignificant amount of time.

Place the fragments in the output area to show the result of running this code.

Outpu	Fragments
Place here	interrupted
Place here	ran
Place here	started
Place here	interrupting
Place here	ended
Place here	InterruptedException:
	(no more output)

Done

Answer:

Give

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

Assume that sleep(n) executes in exactly n milliseconds, and all other code executes in an insignificant amount of time.

Place the fragments in the output area to show the result of running this code.

Outpu	Fragments
started	interrupted
ran	ran
interrupting	started
ended	interrupting
(no more output)	ended
	InterruptedException:
	(no more output)

Done

118. Given that Triangle implements Runnable, and:

```
31.  void go() throws Exception {
32.    Thread t = new Thread(new Triangle());
33.    t.start();
34.    for(int x = 1; x < 100000; x++) {
35.      //insert code here
36.      if(x%100 == 0) System.out.print("g");
37.  } }
38.  public void run() {
39.    try {
40.      for(int x = 1; x < 100000; x++) {
41.        // insert the same code here
42.        if(x%100 == 0) System.out.print("t");
43.      }
44.    } catch (Exception e) { }
```

45. }

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();

Answer: CD

119. 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.

Answer: E

120. Given:

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

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

A. declare reset() using the synchronized keyword

B. declare getName() using the synchronized keyword

C. declare getCount() using the synchronized keyword

D. declare the constructor using the synchronized keyword

E. declare increment() using the synchronized keyword

Answer: ACE

121. Given:

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

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

A. move the line 12 print statement into the foo() method

B. change line 7 to public synchronized void go() {

C. change the variable declaration on line 2 to private volatile int x;

D. wrap the code inside the foo() method with a synchronized(this) block

E. wrap the for loop code inside the go() method with a synchronized block synchronized(this) { // for loop code here }

Answer: AD

122. Given:

```
1. public class TestSeven extends Thread {
2.    private static int x;
3.    public synchronized void doThings() {
4.       int current = x;
5.       current++;
6.       x = current;
7.    }
8.    public void run() {
9.          doThings();
10.   }
11.}
```

Which statement is true?

A. Compilation fails.

B. An exception is thrown at runtime.

C. Synchronizing the run() method would make the class thread-safe.

D. The data in variable "x" are protected from concurrent access problems.

E. Declaring the doThings() method as static would make the class thread-safe.

F. Wrapping the statements within doThings() in a synchronized(new Object()) { } block would make the class thread-safe.

Answer: E

123. Given:

```
11. public class PingPong implements Runnable {
12.     synchronized void hit(long n) {
13.         for(int i = 1; i < 3; i++)
14.             System.out.print(n + "-" + i + " ");
15.     }
16.     public static void main(String[] args) {
```

```
17.         new Thread(new PingPong()).start();
18.         new Thread(new PingPong()).start();
19.     }
20.     public void run() {
21.         hit(Thread.currentThread().getId());
22.     }
23. }
```

Which two statements are true? (Choose two.)

A. The output could be 8-1 7-2 8-2 7-1

B. The output could be 7-1 7-2 8-1 6-1

C. The output could be 8-1 7-1 7-2 8-2

D. The output could be 8-1 8-2 7-1 7-2

Answer: CD

124. 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().

Answer: AF

125. Given:

```
1. package test;
2.
3. class Target {
4.     public String name = "hello";
5. }
```

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

A. any class

B. only the Target class

C. any class in the test package

D. any class that extends Target

Answer: C

126. Given:

```
1.  public class Target {
2.      private int i = 0;
3.      public int addOne(){
4.          return ++i;
5.      }
6.  }
```

And:

```
1.  public class Client {
2.      public static void main(String[] args){
3.          System.out.println(new Target().addOne());
4.      }
5.  }
```

Which change can you make to Target without affecting Client?

A. Line 4 of class Target can be changed to return i++;

B. Line 2 of class Target can be changed to private int i = 1;

C. Line 3 of class Target can be changed to private int addOne(){

D. Line 2 of class Target can be changed to private Integer i = 0;

Answer: D

127. Click Next or More to continue.

Given:

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

and

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

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

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

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

Code Fragments

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

Done

Answer:

Given:

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

and

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

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

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

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

Code Fragments

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

Done

128. Given:

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

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

A. It can be any class.

B. No class has access to base.

C. The class must belong to the geometry package.

D. The class must be a subclass of the class Hypotenuse.

Answer: C

129. 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

Answer: BC

130. 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

Answer: B

131. 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

Answer: A

132. 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

Answer: E

133. Given:

```
11. class Animal { public String noise() { return "peep"; } }
12. class Dog extends Animal {
13.     public String noise() { return "bark"; }
14. }
15. class Cat extends Animal {
16.     public String noise() { return "meow"; }
17. }
...
30. Animal animal = new Dog();
31. Cat cat = (Cat)animal;
32. System.out.println(cat.noise());
```

What is the result?

A. peep

B. bark

C. meow

D. Compilation fails.

E. An exception is thrown at runtime.

Answer: E

134. Given:

```
11. abstract class Vehicle { public int speed() { return 0; }}
12. class Car extends Vehicle { public int speed() { return 60; }}
13. class RaceCar extends Car { public int speed() { return 150; }}
...
21. RaceCar racer = new RaceCar();
22. Car car = new RaceCar();
23. Vehicle vehicle = new RaceCar();
24. System.out.println(racer.speed() + ", " + car.speed()
25.     + ", " + vehicle.speed());
```

What is the result?

A. 0, 0, 0

B. 150, 60, 0

C. Compilation fails.

D. 150, 150, 150

E. An exception is thrown at runtime.

Answer: D

135. Given:

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

And:

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

26. }

What is the result?

A. The code runs with no output.

B. An exception is thrown at runtime.

C. Compilation fails because of an error in line 20.

D. Compilation fails because of an error in line 21.

E. Compilation fails because of an error in line 23.

F. Compilation fails because of an error in line 25.

Answer: F

136. Click Next or More to continue.

Given:

```
class A {
  String name = "A";
  String getName() {
    return name;
  }
  String greeting(){
    return "class A";
  }
}
class B extends A {
  String name = "B";
  String greeting() {
    return "class B";
  }
}
public class Client {
  public static void main( String[] args ) {
    A a = new A();
    A b = new B();
    System.out.println(a.greeting() + " has name " + a.getName());
    System.out.println(b.greeting() + " has name " + b.getName());
  }
}
```

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

Names

class ____ has name Place here

class ____ has name Place here

A B

Done

Answer:

Given:

```
class A {
  String name = "A";
  String getName() {
    return name;
  }
  String greeting(){
    return "class A";
  }
}
class B extends A {
  String name = "B";
  String greeting() {
    return "class B";
  }
}
public class Client {
  public static void main( String[] args ) {
    A a = new A();
    A b = new B();
    System.out.println(a.greeting() + " has name " + a.getName());
    System.out.println(b.greeting() + " has name " + b.getName());
  }
}
```

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

Names

class A A A B

class B A

Done

137. Given:

```
1.  interface A { public void aMethod(); }
2.  interface B { public void bMethod(); }
3.  interface C extends A,B { public void cMethod(); }
4.  class D implements B {
5.      public void bMethod(){}
6.  }
7.  class E extends D implements C {
8.      public void aMethod(){}
9.      public void bMethod(){}
10.     public void cMethod(){}
11. }
```

What is the result?

A. Compilation fails because of an error in line 3.

B. Compilation fails because of an error in line 7.

C. Compilation fails because of an error in line 9.

D. If you define D e = new E(), then e.bMethod() invokes the version of bMethod() defined in Line 5.

E. If you define D e = (D)(new E()), then e.bMethod() invokes the version of bMethod() defined in Line 5.

F. If you define D e = (D)(new E()), then e.bMethod() invokes the version of bMethod() defined in Line 9.

Answer: F

138. Given:

```
5. class Building { }
6. public class Barn extends Building {
7.     public static void main(String[] args) {
8.         Building build1 = new Building();
9.         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.

Answer: C

139. Given:

```
1. class ClassA {
2.     public int numberOfInstances;
3.     protected ClassA(int numberOfInstances) {
```

```
4.        this.numberOfInstances = numberOfInstances;
5.    }
6. }
7. public class ExtendedA extends ClassA {
8.    private ExtendedA(int numberOfInstances) {
9.        super(numberOfInstances);
10.   }
11.   public static void main(String[] args) {
12.       ExtendedA ext = new ExtendedA(420);
13.       System.out.print(ext.numberOfInstances);
14.   }
15. }
```

Which statement is true?

A. 420 is the output.

B. An exception is thrown at runtime.

C. All constructors must be declared public.

D. Constructors CANNOT use the private modifier.

E. Constructors CANNOT use the protected modifier.

Answer: A

140. Click Next or More to continue.

Replace two of the Modifiers that appear in the Single class to make the code compile.
Note: Three modifiers will not be used and four modifiers in the code will remain unchanged.

Code

```
public class Single {
    private   static   Single instance;
    public    static   Single getInstance() {
        if (instance == null) instance = create();
        return instance;
    }
    private   Single() { }
    protected Single create() { return new Single(); }
}

class SingleSub extends Single {
}
```

Modifiers

final
protected
private
abstract
static

Done

Answer:

Replace two of the Modifiers that appear in the Single class to make the code compile.
Note: Three modifiers will not be used and four modifiers in the code will remain unchanged.

Code

```
public class Single {
    private   static   Single instance;
    public    static   Single getInstance() {
        if (instance == null) instance = create();
        return instance;
    }
    protected Single() { }
    static    Single create() { return new Single(); }
}

class SingleSub extends Single {
}
```

Modifiers

final
protected
private
abstract
static

Done

141. Given:

```
1. public class Base {
2.     public static final String FOO = "foo";
3.     public static void main(String[] args) {
4.         Base b = new Base();
5.         Sub s = new Sub();
6.         System.out.print(Base.FOO);
```

```
7.      System.out.print(Sub.FOO);
8.      System.out.print(b.FOO);
9.      System.out.print(s.FOO);
10.     System.out.print(((Base)s).FOO);
11. } }
12. class Sub extends Base {public static final String FOO="bar";}
```

What is the result?

A. foofoofoofoofoo

B. foobarfoobarbar

C. foobarfoofoofoo

D. foobarfoobarfoo

E. barbarbarbarbar

F. foofoofoobarbar

G. foofoofoobarfoo

Answer: D

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

A. A final method in class X can be abstract if and only if X is abstract.

B. A protected method in class X can be overridden by any subclass of X.

C. A private static method can be called only within other static methods in class X.

D. A non-static public final method in class X can be overridden in any subclass of X.

E. A public static method in class X can be called by a subclass of X without explicitly referencing the class X.

F. A method with the same signature as a private final method in class X can be implemented in a subclass of X.

G. A protected method in class X can be overridden by a subclass of X only if the subclass is in the same package as X.

Answer: BEF

143. Click Next or More to continue.

Insert six modifiers into the code such that it meets all of these requirements:

1. It must be possible to create instances of Alpha and Beta from outside the packages in which they are defined.
2. When an object of type Alpha (or any potential subclass of Alpha) has been created, the instance variable alpha may never be changed.
3. The value of the instance variable alpha must always be "A" for objects of type Alpha.

Code

```
package alpha;
Place here    class Alpha {
    Place here    String alpha;
    Place here    Alpha() { this("A"); }
    Place here    Alpha(String a) { alpha = a; }
}

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

Modifiers

private

protected

public

Done

Answer:

Insert six modifiers into the code such that it meets all of these requirements:

1. It must be possible to create instances of Alpha and Beta from outside the packages in which they are defined.
2. When an object of type Alpha (or any potential subclass of Alpha) has been created, the instance variable alpha may never be changed.
3. The value of the instance variable alpha must always be "A" for objects of type Alpha.

Code

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

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

Modifiers

private

protected

public

Done

144. Given:

Given:

10. public class Pizza {

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

And:

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

And:

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

```
 1. class Pizza {
 2.     java.util.ArrayList toppings;
 3.     public final void addTopping(String topping) {
 4.         toppings.add(topping);
 5.     }
 6. }
```

```
7. public class PepperoniPizza extends Pizza {
8.     public void addTopping(String topping) {
9.         System.out.println("Cannot add Toppings");
10.    }
11.    public static void main(String[] args) {
12.        Pizza pizza = new PepperoniPizza();
13.        pizza.addTopping("Mushrooms");
14.    }
15. }
```

What is the result?

A. Compilation fails.

B. Cannot add Toppings

C. The code runs with no output.

D. A NullPointerException is thrown in Line 4.

Answer: A

145. 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.

Answer: E

146. Given:

```
2. public class Hi {
3.     void m1() { }
4.     protected void() m2 { }
5. }
6. class Lois extends Hi {
7.     // insert code here
8. }
```

Which four code fragments, inserted independently at line 7, will compile? (Choose four.)

A. public void m1() { }

B. protected void m1() { }

C. private void m1() { }

D. void m2() { }

E. public void m2() { }

F. protected void m2() { }

G. private void m2() { }

Answer: ABEF

147. Given:

```
1. class Super {
```

```
2.    private int a;
3.    protected Super(int a) { this.a = a; }
4. }
...
11. class Sub extends Super {
12.    public Sub(int a) { super(a); }
13.    public Sub() { this.a = 5; }
14. }
```

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

A. Change line 2 to:

```
public int a;
```

B. Change line 2 to:

```
protected int a;
```

C. Change line 13 to:

```
public Sub() { this(5); }
```

D. Change line 13 to:

```
public Sub() { super(5); }
```

E. Change line 13 to:

```
public Sub() { super(a); }
```

Answer: CD

148. Click the Exhibit button.

```
1. public class SimpleCalc {
2.    public int value;
3.    public void calculate() { value += 7; }
4. }
```

And:

```
1. public class MultiCalc extends SimpleCalc{
2.    public void calculate() { value -= 3; }
3.    public void calculate(int multiplier) {
```

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

What is the result?

A. Value is: 8

B. Compilation fails.

C. Value is: 12

D. Value is: -12

E. The code runs with no output.

F. An exception is thrown at runtime.

Answer: A

149. 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;

Answer: DE

150. Given:

```
1. public class Blip {
2.     protected int blipvert(int x) { return 0; }
3. }
4. class Vert extends Blip {
5.     // insert code here
6. }
```

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

A. public int blipvert(int x) { return 0; }

B. private int blipvert(int x) { return 0; }

C. private int blipvert(long x) { return 0; }

D. protected long blipvert(int x) { return 0; }

E. protected int blipvert(long x) { return 0; }

F. protected long blipvert(long x) { return 0; }

G. protected long blipvert(int x, int y) { return 0; }

Answer: ACEFG

151. Given:

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

and:

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

and:

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

What is the result?

A. 12 4

B. The code runs with no output.

C. An exception is thrown at runtime.

D. Compilation fails because of an error in line 21.

E. Compilation fails because of an error in line 22.

F. Compilation fails because of an error in line 31.

Answer: E

152. Given:

```
3. class Employee {
4.     String name;    double baseSalary;
5.     Employee(String name, double baseSalary) {
6.         this.name = name;
7.         this.baseSalary = baseSalary;
8.     }
9. }
10. public class SalesPerson extends Employee {
11.     double commission;
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);

Answer: AE

153. Given:

```
31. class Foo {
32.     public int a = 3;
33.     public void addFive() { a += 5;    System.out.print("f "); }
34. }
35. class Bar extends Foo {
36.     public int a = 8;
37.     public void addFive() { this.a += 5;
                                System.out.print("b " ); }
38. }
```

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.

Answer: A

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

A. class Man extends Dog { }

B. class Man implements Dog { }

C. class Man { private BestFriend dog; }

D. class Man { private Dog bestFriend; }

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

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

Answer: D

155. Click Next or More to continue.

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

Implementation Structures

```
class A {
  List<B> b;
}
```

```
class A
extends B.C { }
```

```
class A { }
```

```
class A {
  B c; C c;
}
```

```
class A {
  B b;
}
```

```
class A
implements B,C
{ }
```

```
class A
extends B { }
```

Done

Relations

Car is a Vehicle and Car is a Collectable

Car has a SteeringWhe

Car has Wheels

Mini is a Car

Car is an Object

Answer:

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

Implementation Structures

- Car has Wheels
- class A extends B.C { }
- Car is an Object
- class A { B c; C c; }
- Car has a SteeringWhe
- Car is a Vehicle and Car is a Collectable
- Mini is a Car

Done

Relations

- Car is a Vehicle and Car is a Collectable
- Car has a SteeringWhe
- Car has Wheels
- Mini is a Car
- Car is an Object

156. Click Next or More to continue.

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

Type	Relationship	Type	Relationships	Types
Place here	Place	Anima	is-a	Dog
Fores	Place	Place here	has-a	Side
Rectangl	Place	Place here		Tail
Place here	Place	Programming Book		Square
				Tree
				Book
				Java Book
				Pen

Done

Answer:

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

Type	Relationship	Type	Relationships	Types
Dog	is-a	Anima	is-a	Dog
Fores	has-a	Tree	has-a	Side
Rectangl	has-a	Side		Tail
Java Book	is-a	Programming Book		Square
				Tree
				Book
				Java Book
				Pen

Done

157. Given:

```
10. interface Jumper { public void jump(); }
...
20. class Animal {}
...
30. class Dog extends Animal {
31.     Tail tail;
32. }
...
40. class Beagle extends Dog implements Jumper{
41.     public void jump() {}
42. }
...
50. class Cat implements Jumper{
51.     public void jump() {}
52. }
```

Which three are true? (Choose three.)

A. Cat is-a Animal

B. Cat is-a Jumper

C. Dog is-a Animal

D. Dog is-a Jumper

E. Cat has-a Animal

F. Beagle has-a Tail

G. Beagle has-a Jumper

Answer: BCF

158. 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{ }

Answer: AC

159. Given:

5. class Thingy { Meter m = new Meter(); }

6. class Component { void go() { System.out.print("c"); } }

```
7. class Meter extends Component {
        void go() { System.out.print("m"); } }
8.
9. class DeluxeThingy extends Thingy {
10.     public static void main(String[] args) {
11.         DeluxeThingy dt = new DeluxeThingy();
12.         dt.m.go();
13.         Thingy t = new DeluxeThingy();
14.         t.m.go();
15.     }
16. }
```

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.

Answer: AF

160. Given:

```
11. class Mammal { }
12.
13. class Raccoon extends Mammal {
14.     Mammal m = new Mammal();
15. }
16.
17. class BabyRaccoon extends Mammal { }
```

Which four statements are true? (Choose four.)

A. Raccoon is-a Mammal.

B. Raccoon has-a Mammal.

C. BabyRaccoon is-a Mammal.

D. BabyRaccoon is-a Raccoon.

E. BabyRaccoon has-a Mammal.

F. BabyRaccoon is-a BabyRaccoon.

Answer: ABCF

161. Given:

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

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

A. Set set = new TreeSet();

B. Set set = new HashSet();

C. Set set = new SortedSet();

D. List set = new SortedList();

E. Set set = new LinkedHashSet();

Answer: A

162. Given:

```
1. import java.util.*;
2. public class WrappedString {
3.     private String s;
4.     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.

Answer: D

163. Click Next or More to continue.

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

Statements	Collection Types
allows access to elements by their integer index	java.util.Map
defines the method: V get(Object key)	java.util.Set
is designed for holding elements prior to processing	java.util.List
contains no pair of elements e1 and e2, such that	java.util.Queue

Done

Answer:

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

Statements	Collection Types
java.util.List	java.util.Map
java.util.Map	java.util.Set
java.util.Queue	java.util.List
java.util.Set	java.util.Queue

Done

164. Given:

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

Which method will complete this class?

A. public int compareTo(Object o){/*more code here*/}

B. public int compareTo(Score other){/*more code here*/}

C. public int compare(Score s1,Score s2){/*more code here*/}

D. public int compare(Object o1,Object o2){/*more code here*/}

Answer: B

165. A programmer has an algorithm that requires a java.util.List that provides an efficient implementation

of add(0, object), but does NOT need to support quick random access.

What supports these requirements?

A. java.util.Queue

B. java.util.ArrayList

C. java.util.LinearList

D. java.util.LinkedList

Answer: D

166. 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

Answer: D

167. Given:

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

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

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

A. return super.hashCode();

B. return name.hashCode() + age * 7;

C. return name.hashCode() + comment.hashCode() / 2;

D. return name.hashCode() + comment.hashCode() / 2 - age * 3;

Answer: B

168. Given:

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

Which statement is true?

A. The time to find the value from HashMap with a Person key depends on the size of the map.

B. Deleting a Person key from a HashMap will delete all map entries for all keys of type Person.

C. Inserting a second Person object into a HashSet will cause the first Person object to be removed as a duplicate.

D. The time to determine whether a Person object is contained in a HashSet is constant and does NOT depend on the size of the map.

Answer: A

169. Given:

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

Which statement is true?

A. Compilation fails because the hashCode method is not overridden.

B. A HashSet could contain multiple Person objects with the same name.

C. All Person objects will have the same hash code because the hashCode method is not overridden.

D. If a HashSet contains more than one Person object with name="Fred", then removing another Person, also with name="Fred", will remove them all.

Answer: B

170. Given:

```
3. import java.util.*;
4. public class Mapit {
5.     public static void main(String[] args) {
```

```
6.      Set<Integer> set = new HashSet<Integer>();
7.      Integer i1 = 45;
8.      Integer i2 = 46;
9.      set.add(i1);
10.      set.add(i1);
11.      set.add(i2);        System.out.print(set.size() + " ");
12.      set.remove(i1);     System.out.print(set.size() + " ");
13.      i2 = 47;
14.      set.remove(i2);     System.out.print(set.size() + " ");
15.    }
16. }
```

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.

Answer: B

171. Click Next or More to continue.

Place the correct description of the compiler output on the code fragments to be inserted at lines 4 and 5. The same compiler output may be used more than once.

```
1. import java.util.*;
2. public class X {
3.   public static void main(String[] args) {
4.     // insert code here
5.     // insert code here
6.   }
7.   public static void foo(List<Object> list) {
8.   } }
```

Cod

```
ArrayList<String> x1 = new ArrayList<String>();
foo(x1);
```

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

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

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

Compiler Output

Compilation succeeds.

Compilation fails due to an error in the first statement.

Compilation of the first statement succeeds, but compilation fails due to an error in the

Done

Answer:

Place the correct description of the compiler output on the code fragments to be inserted at lines 4 and 5. The same compiler output may be used more than once.

```
1. import java.util.*;
2. public class X {
3.   public static void main(String[] args) {
4.     // insert code here
5.     // insert code here
6.   }
7.   public static void foo(List<Object> list) {
8.   } }
```

Cod

Compilation of the first statement succeeds, but compilation fails due to an error in the

Compilation fails due to an error in the first statement.

Compilation succeeds.

Compilation succeeds.

Compiler Output

Compilation succeeds.

Compilation fails due to an error in the first statement.

Compilation of the first statement succeeds, but compilation fails due to an error in the

Done

172. Click Next or More to continue.

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

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

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

Code

Set<int>	Set<Integer>	HashSet
Set<Integer,String>	Set<int, String>	Set<String,Integer>
Set<String, int>	Set<String>	NumberNames

String	Integer	int	>
>()	name	value	map

Done

Answer:

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

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

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

Code

Set<int>	Set<Integer>	HashSet
Set<Integer,String>	Set<int, String>	Set<String,Integer>
Set<String, int>	Set<String>	NumberNames

String	Integer	int	>
>()	name	value	map

Done

173. Click Next or More to continue.

Give

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

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

```
1. import java.util.*;
2. public class TestGenericConversion {
3.   public static void main(String[] args) {
4.     Place here
5.     list.add("one");
6.     list.add("two");
7.     Place here
8.   }
9. }
```

Code

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

Done

Answer:

Give

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

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

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

Code

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

Done

174. Given:

```
12. import java.util.*;
13. public class Explorer2 {
14.     public static void main(String[] args) {
15.         TreeSet<Integer> s = new TreeSet<Integer>();
16.         TreeSet<Integer> subs = new TreeSet<Integer>();
17.         for(int i = 606; i < 613; i++)
18.             if(i%2 == 0) s.add(i);
19.         subs = (TreeSet)s.subSet(608, true, 611, true);
20.         s.add(629);
21.         System.out.println(s + " " + subs);
22.     }
23. }
```

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]

Answer: E

175. Given:

```
12. import java.util.*;
13. public class Explorer1 {
14.     public static void main(String[] args) {
15.         TreeSet<Integer> s = new TreeSet<Integer>();
16.         TreeSet<Integer> subs = new TreeSet<Integer>();
17.         for(int i = 606; i < 613; i++)
```

```
18.         if(i%2 == 0) s.add(i);
19.     subs = (TreeSet)s.subSet(608, true, 611, true);
20.     s.add(609);
21.     System.out.println(s + " " + subs);
22.  }
23. }
```

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]

Answer: F

176. Given:

```
12. import java.util.*;
13. public class Explorer3 {
14.     public static void main(String[] args) {
15.         TreeSet<Integer> s = new TreeSet<Integer>();
16.         TreeSet<Integer> subs = new TreeSet<Integer>();
17.         for(int i = 606; i < 613; i++)
18.             if(i%2 == 0) s.add(i);
19.         subs = (TreeSet)s.subSet(608, true, 611, true);
20.         subs.add(629);
21.         System.out.println(s + " " + subs);
22.     }
23. }
```

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]

Answer: B

177. 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)".

Answer: ACF

178. Click Next or More to continue.

Given:

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

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

Method Calls		Result
m1(listA);	m2(listA);	Does not compile.
m1(listB);	m2(listB);	Compiles and runs without
m1(listO);	m2(listO);	An exception is thrown at

Done

Answer:

Given:

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

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

Method Calls		Result
Compiles and runs without	Compiles and runs without	Does not compile.
Compiles and runs without	Does not compile.	Compiles and runs without
Does not compile.	Does not compile.	An exception is thrown at

Done

179. Click Next or More to continue.

Given the class definitions:

class Animal { }
class Dog extends Animal { }

and the code:

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

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

takeList() Method Definition

- public void takeList(ArrayList list) { }
- public void takeList(ArrayList<Animal> list) { }
- public void takeList(ArrayList<? extends Animal> list) { }
- public void takeList(ArrayList<?> list) { }
- public void takeList(ArrayList<Object> list) { }

Compilation Result

- Compilation succeeds.
- Compilation fails.

Done

Answer:

Given the class definitions:

class Animal { }
class Dog extends Animal { }

and the code:

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

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

takeList() Method Definition

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

Compilation Result

- Compilation succeeds.
- Compilation fails.

Done

180. Click Next or More to continue.

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

Class

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

Code Options

Gen<T>

Gen<?>

Gen

?

T

Done

Answer:

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

Class

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

Code Options

Gen<T>

Gen<?>

Gen

?

T

Done

181. Given:

11. // insert code here

12. private N min, max;

13. public N getMin() { return min; }

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

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

A. public class MinMax<?> {

B. public class MinMax<? extends Number> {

C. public class MinMax<N extends Object> {

D. public class MinMax<N extends Number> {

E. public class MinMax<? extends Object> {

F. public class MinMax<N extends Integer> {

Answer: DF

182. A programmer must create a generic class MinMax and the type parameter of MinMax must implement Comparable.

Which implementation of MinMax will compile?

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

B.
```
class MinMax<E implements Comparable<E>> {
   E min = null;
   E max = null;
   public MinMax() {}
```

```
   public void put(E value) { /* store min or max */ }
}
```

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

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

Answer: A

183. 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>();

Answer: BCD

184. Given:

```
3. import java.util.*;
4. public class Hancock {
5.     // insert code here
6.         list.add("foo");
7.     }
8. }
```

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) {

Answer: BC

185. Given:

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

and:

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

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

What is the result?

A. Tea

B. Coffee

C. Coffee

Tea

D. Compilation fails.

E. The code runs with no output.

F. An exception is thrown at runtime.

Answer: B

186. 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);

Answer: B

187. 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.

Answer: C

188. Given:

```
5. import java.util.*;
6. public class SortOf {
7.     public static void main(String[] args) {
8.         ArrayList<Integer> a = new ArrayList<Integer>();
9.         a.add(1);    a.add(5);    a.add(3);
11.         Collections.sort(a);
12.         a.add(2);
13.         Collections.reverse(a);
14.         System.out.println(a);
15.     }
16. }
```

What is the result?

A. [1, 2, 3, 5]

B. [2, 1, 3, 5]

C. [2, 5, 3, 1]

D. [5, 3, 2, 1]

E. [1, 3, 5, 2]

F. Compilation fails.

G. An exception is thrown at runtime.

Answer: C

189. 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.

Answer: AC

190. Given classes defined in two different files:

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

What is required at line 5 in class SomeApp to use the process method of BitUtils?

A. process(bytes);

B. BitUtils.process(bytes);

C. util.BitUtils.process(bytes);

D. SomeApp cannot use methods in BitUtils.

E. import util.BitUtils.*; process(bytes);

Answer: C

191. Given classes defined in two different files:

```
1. package util;
2. public class BitUtils {
3.     private static void process(byte[] b) {}
4. }
```

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

What is required at line 5 in class SomeApp to use the process method of BitUtils?

A. process(bytes);

B. BitUtils.process(bytes);

C. app.BitUtils.process(bytes);

D. util.BitUtils.process(bytes);

E. import util.BitUtils.*; process(bytes);

F. SomeApp cannot use the process method in BitUtils.

Answer: F

192. Given a class Repetition:

```
1.  package utils;
2.
3.  public class Repetition {
```

```
4.      public static String twice(String s) { return s + s; }
5.  }
```

and given another class Demo:

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

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

A. import utils.*;

B. static import utils.*;

C. import utils.Repetition.*;

D. static import utils.Repetition.*;

E. import utils.Repetition.twice();

F. import static utils.Repetition.twice;

G. static import utils.Repetition.twice;

Answer: F

193. Given:

```
11. interface DeclareStuff {
12.     public static final int EASY = 3;
13.     void doStuff(int t); }
14. public class TestDeclare implements DeclareStuff {
15.     public static void main(String [] args) {
16.         int x = 5;
17.         new TestDeclare().doStuff(++x);
18.     }
19.     void doStuff(int s) {
```

```
20.         s += EASY + ++s;
21.         System.out.println("s " + s);
22.     }
23. }
```

What is the result?

A. s 14

B. s 16

C. s 10

D. Compilation fails.

E. An exception is thrown at runtime.

Answer: D

194. Given:

```
 1. interface DoStuff2 {
 2.     float getRange(int low, int high); }
 3.
 4. interface DoMore {
 5.     float getAvg(int a, int b, int c); }
 6.
 7. abstract class DoAbstract implements DoStuff2, DoMore { }
 8.
 9. class DoStuff implements DoStuff2 {
10.     public float getRange(int x, int y) { return 3.14f; }    }
11.
12. interface DoAll extends DoMore {
13.     float getAvg(int a, int b, int c, int d); }
```

What is the result?

A. The file will compile without error.

B. Compilation fails. Only line 7 contains an error.

C. Compilation fails. Only line 12 contains an error.

D. Compilation fails. Only line 13 contains an error.

E. Compilation fails. Only lines 7 and 12 contain errors.

F. Compilation fails. Only lines 7 and 13 contain errors.

G. Compilation fails. Lines 7, 12, and 13 contain errors.

Answer: A

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

```
1. package animals.mammals;
2.
3. public class GrizzlyBear extends Bear {
4.     void hunt() {
5.         Salmon s = findSalmon();
6.         s.consume();
7.     }
8. }
```

```
1. package animals.fish;
2.
3. public class Salmon extends Fish {
4.     public void consume() { /* do stuff */ }
5. }
```

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

Answer: B

196. Given a correctly compiled class whose source code is:

```
1. package com.sun.sjcp;
```

```
2. public class Commander {
3.     public static void main(String[] args) {
4.         // more code here
5.     }
6. }
```

Assume that the class file is located in /foo/com/sun/sjcp/, the current directory is /foo/, and that the classpath contains "." (current directory).

Which command line correctly runs Commander?

A. java Commander

B. java com.sun.sjcp.Commander

C. java com/sun/sjcp/Commander

D. java -cp com.sun.sjcp Commander

E. java -cp com/sun/sjcp Commander

Answer: B

197. 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.

Answer: C

198. Given:

```
3. interface Animal { void makeNoise(); }
4. class Horse implements Animal {
5.     Long weight = 1200L;
6.     public void makeNoise() { System.out.println("whinny"); }
7. }
8. public class Icelandic extends Horse {
9.     public void makeNoise() { System.out.println("vinny"); }
10.     public static void main(String[] args) {
11.         Icelandic i1 = new Icelandic();
12.         Icelandic i2 = new Icelandic();
13.         Icelandic i3 = new Icelandic();
14.         i3 = i1; i1 = i2; i2 = null; i3 = i1;
15.     }
16. }
```

When line 15 is reached, how many objects are eligible for the garbage collector?

A. 0

B. 1

C. 2

D. 3

E. 4

F. 6

Answer: E

199. Given:

```
11. public class ItemTest {
12.     private final int id;
13.     public ItemTest(int id) { this.id = id; }
14.     public void updateId(int newId) { id = newId; }
```

```
15.
16.     public static void main(String[] args) {
17.         ItemTest fa = new ItemTest(42);
18.         fa.updateId(69);
19.         System.out.println(fa.id);
20.     }
21. }
```

What is the result?

A. Compilation fails.

B. An exception is thrown at runtime.

C. The attribute id in the ItemTest object remains unchanged.

D. The attribute id in the ItemTest object is modified to the new value.

E. A new ItemTest object is created with the preferred value in the id attribute.

Answer: A

200. Given:

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

What is the result?

A. Compilation fails.

B. An exception is thrown at runtime.

C. doStuff x = 6 main x = 6

D. doStuff x = 5 main x = 5

E. doStuff x = 5 main x = 6

F. doStuff x = 6 main x = 5

Answer: D

201. Click Next or More to continue.

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

```
Place here        update(int quantity, int adjust) {
  Place here

public void callUpdate() {
 int quant = 100;
  Place here

 System.out.println("The quantity is " + quant);
}
```

Code Fragments

public int	quantity = quantity	update(quant, 320);
public void	quant = update(quant, 320);	quantity = quantity + adjust; return quantity;

Done

Answer:

Place code fragments into position so the output is:

The quantity is 420

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


public void callUpdate() {
 int quant = 100;
    quant = update(quant, 320);

 System.out.println("The quantity is " + quant);
}
```

Code Fragments

public int	quantity = quantity	update(quant, 320);
public void	quant = update(quant, 320);	quantity = quantity + adjust; return quantity;

Done

202. Click the Exhibit button.

```
10. class Inner {
11.     private int x;
12.     public void setX( int x ) { this.x = x; }
13.     public int getX() { return x; }
14. }
15.
16. class Outer {
17.     private Inner y;
18.     public void setY( Inner y ) { this.y = y; }
19.     public Inner getY() { return y; }
20. }
21.
22. public class Gamma {
23.     public static void main( String[] args ) {
24.         Outer o = new Outer();
```

```
25.         Inner i = new Inner();
26.         int n = 10;
27.         i.setX( n );
28.         o.setY( i );
29.         // insert code here
30.  System.out.println( o.getY().getX() );
31.     }
32. }
```

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

A. n = 100;

B. i.setX(100);

C. o.getY().setX(100);

D. i = new Inner(); i.setX(100);

E. o.setY(i); i = new Inner(); i.setX(100);

F. i = new Inner(); i.setX(100); o.setY(i);

Answer: BCF

203. Click the Exhibit button.

```
10. class Foo {
11.    private int x;
12.    public Foo( int x ) { this.x = x; }
13.    public void setX( int x ) { this.x = x; }
14.    public int getX() { return x; }
15. }
16.
17. public class Gamma {
18.
19.     static Foo fooBar( Foo foo ) {
20.         foo = new Foo( 100 );
21.         return foo;
```

```
22.     }
23.
24.     public static void main( String[] args ) {
25.  Foo foo = new Foo( 300 );
26.  System.out.print( foo.getX() + "-" );
27.
28.  Foo fooFoo = fooBar( foo );
29.  System.out.print( foo.getX() + "-" );
30.  System.out.print( fooFoo.getX() + "-" );
31.
32.  foo = fooBar( fooFoo );
33.  System.out.print( foo.getX() + "-" );
34.  System.out.print( fooFoo.getX() );
35.     }
36. }
```

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

A. 300-100-100-100-100

B. 300-300-100-100-100

C. 300-300-300-100-100

D. 300-300-300-300-100

Answer: B

204. Given:

```
5. class Payload {
6.     private int weight;
7.     public Payload (int w) { weight = w; }
8.     public void setWeight(int w) { weight = w; }
9.     public String toString() { return Integer.toString(weight); }
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. } }
```

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);

Answer: A

205. Given:

```
3. public class Batman {
4.     int squares = 81;
5.     public static void main(String[] args) {
6.         new Batman().go();
7.     }
8.     void go() {
9.         incr(++squares);
10.         System.out.println(squares);
11.     }
12.     void incr(int squares) { squares += 10; }
13. }
```

What is the result?

A. 81

B. 82

C. 91

D. 92

E. Compilation fails.

F. An exception is thrown at runtime.

Answer: B

206. Given:

```
11. public void genNumbers() {
12.     ArrayList numbers = new ArrayList();
13.     for (int i=0; i<10; i++) {
14.         int value = i * ((int) Math.random());
15.         Integer intObj = new Integer(value);
16.         numbers.add(intObj);
17.     }
18.     System.out.println(numbers);
19. }
```

Which line of code marks the earliest point that an object referenced by intObj becomes a candidate for garbage collection?

A. Line 16

B. Line 17

C. Line 18

D. Line 19

E. The object is NOT a candidate for garbage collection.

Answer: D

207. Given:

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

```
15.
16. class Boochy {
17.     Snoochy snooch;
18.     public Boochy(Snoochy s) { snooch = s; }
19. }
```

And the statements:

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

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

A. None of these objects are eligible for garbage collection.

B. Only the object referenced by booch is eligible for garbage collection.

C. Only the object referenced by snoog is eligible for garbage collection.

D. Only the object referenced by snooch is eligible for garbage collection.

E. The objects referenced by snooch and booch are eligible for garbage collection.

Answer: E

Testinside

Testinside.com was founded in 2006. The safer,easier way to help you pass any IT Certification exams . We provide high quality IT Certification exams practice questions and answers(Q&A). Especially Adobe, Apple, Citrix, Comptia, EMC, HP, Juniper, LPI, Nortel, Oracle, SUN, Vmware and so on. And help you pass any IT Certification exams at the first try.

You can reach us at any of the email addresses listed below.

English Customer: Sales(at)TestInside.Com
TaiWan&HK Customer: Salestw(at)TestInside.Com
Chinese Customer: Salescn(at)TestInside.com

English Version http://www.testinside.com
Chinese (Traditional) http:// www.testinside.net
Chinese (Simplified) http:// www.testinside.cn