

## Soal Code A

### Q: 1 Given:

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

### Q: 2 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);

### Q: 3 Which statement is true about the set variable on line 12?

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

- A. The set variable contains all six elements from the coll collection, and the order is guaranteed to be preserved.
- B. The set variable contains only three elements from the coll collection, and the order is guaranteed to be preserved.
- C. The set variable contains all six elements from the coll collection, but the order is NOT guaranteed to be preserved.

- D. The set variable contains only three elements from the coll collection, but the order is NOT guaranteed to be preserved.

### Q: 4 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.

### Q: 5

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 here {
    private Place here object;
    public Place here (Place here object) {
        this.object = object;
    }
    public Place here 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

### Q: 6

### Soal Code A

Given:

```
public void takeList(List<? extends String> list) {  
    // insert code here  
}
```

Place the Compilation Results on each code statement to indicate whether or not that code will compile if inserted into the takeList() method.

#### Code Statements

```
list.add("Foo");
```

```
list = new ArrayList<String>();
```

```
list = new ArrayList<Object>();
```

```
String s = list.get(0);
```

```
Object o = list;
```

#### Compilation Results

Compilation succeeds

Compilation fails

Done

### Q: 7

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.     }  
}
```

#### Code

```
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 second statement.

Done

### Q: 8 Given:

```
1. public class Person {  
2.     private String name;  
3.     public Person(String name) { this.name =  
         name; }  
4.     public boolean equals(Person p) {  
5.         return p.name.equals(this.name);  
6.     }  
7. }
```

Which statement is true?

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

### Q: 9 Given:

```
1. import java.util.*;  
2. public class Old {  
3.     public static Object get0(List list) {  
4.         return list.get(0);  
5.     }  
6. }
```

Which three will compile successfully? (Choose three.)

- A. Object o = Old.get0(new LinkedList());
- B. Object o = Old.get0(new LinkedList<?>());
- C. String s = Old.get0(new LinkedList<String>());
- D. Object o = Old.get0(new LinkedList<Object>());
- E. String s = (String)Old.get0(new LinkedList<String>());

### Q: 10 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();

### Q: 11 Given

```
10. class Foo {  
11.     static void alpha() { /* more code here  
        */ }  
12.     void beta() { /* more code here */ }  
13. }
```

Which two statements are true? (Choose two.)

- A. Foo.beta() is a valid invocation of beta().
- B. Foo.alpha() is a valid invocation of alpha().
- C. Method beta() can directly call method alpha().
- D. Method alpha() can directly call method beta().

### Q: 12

### Soal Code A

Place the Output Options in the Actual Output Sequence to indicate the output from this code:

```
class Alpha {
    public void foo( String... args )
    { System.out.print("Alpha:foo "); }
    public void bar( String a )
    { System.out.print("Alpha:bar "); }
}

public class Beta extends Alpha {
    public void foo( String a )
    { System.out.print("Beta:foo "); }
    public void bar( String a )
    { System.out.print("Beta:bar "); }
    public static void main( String[] argv ) {
        Alpha a = new Beta();
        Beta b = (Beta)a;
        a.foo( "test" ); b.foo( "test" );
        a.bar( "test" ); b.bar( "test" );
    }
}
```

#### Actual Output Sequence

Place here

Place here

Place here

Place here

#### Output Options

Alpha.foo

Alpha.bar

Beta.foo

Beta.bar

Done

### Q: 13 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.

### Q: 14 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 instantiated at line 15.
- D. Line l = new Line(); l.Point p = new l.Point();

### Q: 15

RZQ 12/18/2008

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,
```

```
}
```

### Q: 16 Given:

```
10. package com.sun.scjp;
11. public class Geodetics {
12. public static final double DIAMETER =
12756.32; // kilometers
13. }
```

Which two correctly access the DIAMETER member of the Geodetics class? (Choose two.)

- A. 

```
import com.sun.scjp.Geodetics;
public class TerraCarta {
    public double halfway()
    { return Geodetics.DIAMETER/2.0; }
```
- B. 

```
import static com.sun.scjp.Geodetics;
public class TerraCarta{
    public double halfway() { return
    DIAMETER/2.0; }
```
- C. 

```
import static com.sun.scjp.Geodetics.*;
public class TerraCarta {
    public double halfway() { return
    DIAMETER/2.0; }
```
- D. 

```
package com.sun.scjp;
public class TerraCarta {
    public double halfway() { return
    DIAMETER/2.0; }
```

### Q: 17 Given:

```
10. public class Bar {
11. static void foo( int... x ) {
12. // insert code here
13. }
14. }
```

Which two code fragments, inserted independently at line 12, will allow the class to compile? (Choose two.)

- A. `foreach( x ) System.out.println(z);`
- B. `for( int z : x ) System.out.println(z);`
- C. `while( x.hasNext() ) System.out.println( x.next() );`

**Soal Code A**

D. for( int i=0; i< x.length; i++ ) System.out.println(x[i]);

**Q: 18 Which statement is true about the classes and interfaces in the exhibit?**

```

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. }
    
```

- 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.

**Q: 19 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.

**Q: 20**

Place the code elements in order so that the resulting Java source file will compile correctly, resulting in a class called com.sun.cert.AddressBook.

| Source File        | Code Element                      |
|--------------------|-----------------------------------|
| 1st                | package com.sun.cert;             |
| 2nd                | package com.sun.cert.*;           |
| 3rd                | import java.util.*;               |
|                    | import java.*;                    |
| ArrayList entries; | public class AddressBook {        |
| }                  | public static class AddressBook { |

**Q: 21**

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       |

**Q: 22 Which two classes correctly implement both the java.lang.Runnable and the java.lang.Cloneable interfaces? (Choose two.)**

- A. 

```
public class Session implements Runnable, Cloneable { public void run(); public Object clone(); }
```
- B. 

```
public class Session extends Runnable, Cloneable { public void run() { /* do something */ } public Object clone() { /* make a copy */ } }
```
- C. 

```
public class Session implements Runnable, Cloneable { public void run() { /* do something */ } public Object clone() { /* make a copy */ } }
```
- D. 

```
public abstract class Session implements Runnable, Cloneable { public void run() { /* do something */ } public Object clone() { /*make a copy */ } }
```

### Soal Code A

```
E. public class Session
    implements Runnable, implements Cloneable
    {
    public void run() { /* do something */ }
    public Object clone() { /* make a copy */ }
    }
```

### Q: 23 What is the result?

```
1. class Computation extends Thread {
2.
3.     private int num;
4.     private boolean isComplete;
5.     private int result;
6.
7.     public Computation(int num) { this.num
= num; }
8.
9.     public synchronized void run() {
10.         result = num * 2;
11.         isComplete = true;
12.         notify();
13.     }
14.
15.     public synchronized int getResult() {
16.         while (!isComplete) {
17.             try {
18.                 wait();
19.             } catch (InterruptedException e)
{}
20.         }
21.         return result;
22.     }
23.
24.     public static void main(String[] args)
{
25.         Computation[] computations = new
Computation[4];
26.         for (int i = 0; i <
computations.length; i++) {
27.             computations[i] = new
Computation(i);
28.             computations[i].start();
29.         }
30.         for (Computation c : computations)
31.             System.out.print(c.getResult() +
");
32.     }
33. }
```

- A. The code will deadlock.
- B. The code may run with no output.
- C. An exception is thrown at runtime.
- D. The code may run with output "0 6".
- E. The code may run with output "2 0 6 4".
- F. The code may run with output "0 2 4 6".

### Q: 24 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.
- C. java.lang.RuntimeException: Problem
- D. End of method.
- E. java.lang.RuntimeException: Problem
- F. run.
- G. java.lang.RuntimeException: Problem
- H. End of method.

### Q: 25 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.

### Q: 26 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".

### Q: 27 Given:

```
public class NamedCounter {
    private final String name;
    private int count;
    public NamedCounter(String name) { this.name
= name; }
    public String getName() { return name; }
}
```

### Soal Code A

```
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

### Q: 28 Given:

```
7. void waitForSignal() {
8. Object obj = new Object();
9. synchronized (Thread.currentThread()) {
10. obj.wait();
11. obj.notify();
12. }
13. }
```

Which statement is true?

- A. This code may throw an InterruptedException.
- B. This code may throw an IllegalStateException.
- C. This code may throw a TimeoutException after ten minutes.
- D. This code will not compile unless "obj.wait()" is replaced with "((Thread) obj).wait()".
- E. Reversing the order of obj.wait() and obj.notify() may cause this method to complete normally.
- F. A call to notify() or notifyAll() from another thread may cause this method to complete normally.

### Q: 29 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();
```

### Q: 30 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".

### Q: 31 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.

### Q: 32 Given:

```
8. public class test {
9. public static void main(String [] a) {
10. assert a.length == 1;
11. }
12. }
```

Which two will produce an AssertionError? (Choose two.)

- A. java test
- B. java -ea test
- C. java test file1
- D. java -ea test file1
- E. java -ea test file1 file2
- F. java -ea:test test file1

Answer: B, E

### Q: 33 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;

### Soal Code A

- B.  $f = (\Delta)x$ ;
- C.  $f = (\text{Alpha})x$ ;
- D.  $b = (\text{Beta})(\text{Alpha})x$ ;

### Q: 34 Given:

```
11. public static Collection get() {
12.     Collection sorted = new LinkedList();
13.     sorted.add("B");     sorted.add("C");
14.     sorted.add("A");
15.     return sorted;
16. }
17. public static void main(String[] args) {
18.     for (Object obj: get()) {
19.         System.out.print(obj + ", ");
20.     }
```

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.

### Q: 35 Given:

```
84. try {
85.     ResourceConnection con =
resourceFactory.getConnection();
86.     Results r = con.query("GET INFO FROM
CUSTOMER");
87.     info = r.getData();
88.     con.close();
89. } catch (ResourceException re) {
90.     errorLog.write(re.getMessage());
91. }
92. return info;
```

Which statement is true if a ResourceException is thrown on line 86?

- A. Line 92 will not execute.
- B. The connection will not be retrieved in line 85.
- C. The resource connection will not be closed on line 88.
- D. The enclosing method will throw an exception to its caller.

### Q: 36 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.

### Q: 37

Place the correct Code in the Code Sample to achieve the expected results.

### Expected Results

Output: 1 2 4 8 16 32

### Code Sample

```
int [] y = { 1, 2, 4, 8, 16, 32 };
System.out.print("Output: ");
```

Place here

```
System.out.print(x);
System.out.print(" ");
}
```

### Code

```
for(int x : y) { } for(int x = y[]) { }
foreach (y as x) { } foreach (int x : y) { }
for(int x=1; x=y[]; x++) { }
```

### Q: 38 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"); } }
```

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.

### Q: 39 Given a method that must ensure that its parameter is not null:

```
11. public void someMethod(Object value) {
12.     // check for null value
...
20. System.out.println(value.getClass());
21. }
```

What, inserted at line 12, is the appropriate way to handle a null value?

- A. `assert value == null;`
- B. `assert value != null, "value is null";`
- C. 

```
if (value == null) {
    throw new
    AssertionError("value is
    null");
}
```
- D. 

```
if (value == null) {
    throw new
    IllegalArgumentException("value is
    null");
}
```

## Soal Code A

### Q: 40 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) {
18. System.out.print("exception "); }
19. 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.

### Q: 41 Given:

```
11. Float pi = new Float(3.14f);
12. if (pi > 3) {
13. System.out.print("pi is bigger than 3.
14. ");
15. }
16. else {
17. System.out.print("pi is not bigger than
18. 3. ");
19. }
20. finally {
21. System.out.println("Have a nice day.");
22. }
```

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.

### Q: 42

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 PrintWriter()     new BufferedReader()  
 new BufferedWriter()     new FileWriter()     new PrintWriter()

### Q: 43 Given:

```
11. String test = "This is a test";
12. String[] tokens = test.split("\s");
13. System.out.println(tokens.length);
```

What is the result?

- A. 0
- B. 1
- C. 4
- D. Compilation fails.

E. An exception is thrown at runtime.

### Q: 44 Given:

```
11. public class Yikes {
12.
13.     public static void go(Long n)
14.     {System.out.println("Long ");}
15.     public static void go(Short n)
16.     {System.out.println("Short ");}
17.     public static void go(int n)
18.     {System.out.println("int ");}
19.     public static void main(String [] args) {
20.     short y = 6;
21.     long z = 7;
22.     go(y);
23.     go(z);
24. }
```

What is the result?

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

### Q: 45 Given:

```
12. System.out.format("Pi is approximately
13. %d.", Math.PI);
```

What is the result?

- A. Compilation fails.
- B. Pi is approximately 3.
- C. Pi is approximately 3.141593.
- D. An exception is thrown at runtime.

### Q: 46 Given:

```
33. Date d = new Date(0);
34. String ds = "December 15, 2004";
35. // insert code here
36. try {
37.     d = df.parse(ds);
38. }
39. catch(ParseException e) {
40.     System.out.println("Unable to parse " +
41.     ds);
42. }
43. // insert code here too
```

What creates the appropriate DateFormat object and adds a day to the Date object?

- A. 

```
35.     DateFormat df =
36.     DateFormat.getDateFormat();
37.     d.setTime( (60 * 60 * 24) +
38.     d.getTime());
```
- B. 

```
35.     DateFormat df =
36.     DateFormat.getDateInstance();
37.     d.setTime( (1000 * 60 * 60 * 24) +
38.     d.getTime());
```
- C. 

```
35.     DateFormat df =
36.     DateFormat.getDateFormat();
37.     d.setLocalTime( (1000*60*60*24) +
38.     d.getLocalTime());
```
- D. 

```
35.     DateFormat df =
36.     DateFormat.getDateInstance();
37.     d.setLocalTime( (60 * 60 * 24) +
38.     d.getLocalTime());
```

### Q: 47 Given:



### Soal Code A

```
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.

### Q: 48 Given:

```
12. import java.io.*;
13.     public class Forest implements
Serializable {
14.     private Tree tree = new Tree();
15.     public static void main(String [] args) {
16.     Forest f = new Forest();
17.     try {
18.         FileOutputStream fs = new
FileOutputStream("Forest.ser");
19.         ObjectOutputStream os = new
ObjectOutputStream(fs);
20.     os.writeObject(f); os.close();
21.     } catch (Exception ex) {
ex.printStackTrace(); }
22. } }
23.
24. class Tree { }
```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. An instance of Forest is serialized.
- D. An instance of Forest and an instance of Tree are both serialized.

### Q: 49

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.readLine())

&& buffReader.hasNext()

(temp = buffReader.nextLine())

(IOException e) {

!= null

(FileNotFoundException e) {

Done

**Q: 50 Assuming that the serializeBanana() and the deserializeBanana() methods will correctly use Java serialization and given:**

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

What is the result?

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

### Q: 51 Given this method in a class:

```
21. public String toString() {
22.     StringBuffer buffer = new StringBuffer();
23.     buffer.append('<');
24.     buffer.append(this.name);
25.     buffer.append('>');
26.     return buffer.toString();
27. }
```

Which statement is true?

- A. This code is NOT thread-safe.
- B. The programmer can replace StringBuffer with StringBuilder with no other changes.
- C. This code will perform poorly. For better performance, the code should be rewritten:  
return "<" + this.name + ">";
- D. This code will perform well and converting the code to use StringBuilder will not enhance the performance.

### Q: 52

**Soal Code A**

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();
        B 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

class  has name

**Q: 53**

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

| Implementation Structures           | Relations                                       |                            |
|-------------------------------------|---|----------------------------|
| class A {<br>List<B> b;<br>}        | Car is a Vehicle<br>and<br>Car is a Collectable |                            |
| class A<br>extends B,C { }          |   |                            |
| class A { }                         |   | Car has a<br>SteeringWheel |
| class A {<br>B b;<br>}              |   | Car has Wheels             |
| class A {<br>B c; C c;<br>}         |   | Mini is a Car              |
| class A<br>implements B,C<br>{ }    |   | Car is an Object           |
| class A<br>extends B { }            |   |                            |
| <input type="button" value="Done"/> |   |                            |

**Q: 54 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.

**Q: 55 Which two statements are true about has-a and is-a relationships? (Choose two.)**

- A. Inheritance represents an is-a relationship.
- B. Inheritance represents a has-a relationship.
- C. Interfaces must be used when creating a has-a relationship.
- D. Instance variables can be used when creating a has-a relationship.

**Q: 56 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.

**Q: 57 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); }

## Soal Code A

### Q: 58 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.

### Q: 59 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.

### Q: 60 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?

RZQ 12/18/2008

- A. fofofofofofofo
- B. foobarfoobarbar
- C. foobarfofofofofo
- D. foobarfoobarfoo
- E. barbarbarbarbar
- F. fofofofoobarbar
- G. fofofofoobarfoo