

**Soal Code C**

**Q: 121 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;

**Q: 122 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; }`

**Q: 123**

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

<pre> class Alpha {     public void bar( int... x ) {}     public void bar( int x ) {} }  public class Beta extends Alpha {     _____     _____     _____ }         </pre>	<p><b>Methods</b></p> <ul style="list-style-type: none"> <li><code>private void bar( int x ) {}</code></li> <li><code>public void bar( int x ) {}</code></li> <li><code>public int bar( String x ) { return 1; }</code></li> <li><code>public Alpha bar( int x ) {}</code></li> <li><code>public void bar( int x, int y ) {}</code></li> <li><code>public int bar( int x ) { return x; }</code></li> </ul>
--	--

**Q: 124**

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
    _____ {
    _____
}
    
```

**Code Fragments**

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

**Q: 125 What is the result?**

```

11. public class Bootchy {
12.     int bootch;
13.     String snootch;
14.
15.     public Bootchy() {
16.         this("snootchy");
17.         System.out.print("first ");
18.     }
19.
20.     public Bootchy(String snootch) {
21.         this(420, "snootchy");
22.         System.out.print("second ");
23.     }
24.
25.     public Bootchy(int bootch, String
    snootch) {
26.         this.bootch = bootch;
27.         this.snootch = snootch;
28.         System.out.print("third ");
29.     }
30.
31.     public static void main(String[] args)
    {
32.         Bootchy b = new Bootchy();
33.         System.out.print(b.snootch + " " +
    b.bootch);
34.     }
35. }
    
```

- A. snootchy 420 third second first
- B. snootchy 420 first second third
- C. first second third snootchy 420
- D. third second first snootchy 420
- E. third first second snootchy 420
- F. first second first third snootchy 420

**Q: 126 A JavaBeans component has the following field:**

```
11. private boolean enabled;
```

Which two pairs of method declarations follow the JavaBeans standard for accessing this field? (Choose two.)

- A. `public void setEnabled( boolean enabled )`  
`public boolean getEnabled()`
- B. `public void setEnabled( boolean enabled )`  
`public void isEnabled()`
- C. `public void setEnabled( boolean enabled )`  
`public boolean isEnabled()`

### Soal Code C

```
D. public boolean setEnabled( boolean
    enabled )
    public boolean getEnabled()
```

### Q: 127 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)
15.         System.out.print(i + " ");
16. }
```

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.

### Q: 128 Given:

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

```
1. public class Threads1 {
2.     int x = 0;
3.     public class Runner implements Runnable
4.     {
5.         public void run() {
6.             int current = 0;
7.             for(int i = 0; i < 4; i++) {
8.                 current = x;
9.                 System.out.print(current + " ");
10.                x = current + 2;
11.            }
12.        }
13.    }
14.     public static void main(String[] args) {
15.         new Threads1().go();
16.     }
17.     public void go() {
18.         Runnable r1 = new Runner();
19.         new Thread(r1).start();
20.         new Thread(r1).start();
21.     }
22. }
23. }
```

- A. 0, 2, 4, 4, 6, 8, 10, 6,
- B. 0, 2, 4, 6, 8, 10, 2, 4,
- C. 0, 2, 4, 6, 8, 10, 12, 14,

- D. 0, 0, 2, 2, 4, 4, 6, 6, 8, 8, 10, 10, 12, 12, 14, 14,
- E. 0, 2, 4, 6, 8, 10, 12, 14, 0, 2, 4, 6, 8, 10, 12, 14,

### Q: 130 Which three will compile and run without exception? (Choose three.)

- A. private synchronized Object o;
- B. void go() { synchronized() { /\* code here \*/ } }
- C. public synchronized void go() { /\* code here \*/ }
- D. private synchronized(this) void go() { /\* code here \*/ }
- E. void go() { synchronized(Object.class) { /\* code here \*/ } }
- F. void go() { Object o = new Object(); synchronized(o) { /\* code here \*/ } }

### Q: 131 Given:

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

### Q: 132 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.

### Q: 133 What is the result?

### Soal Code C

```

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: 134 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: 135 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: 136 What is the result?

```

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. }

```

- 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.

#### Q: 137 Given:

```

10. interface A { public int getValue(); }
11. class B implements A {
12.     public int getValue() { return 1; }
13. }
14. class C extends B {
15.     // insert code here
16. }

```

Which three code fragments, inserted individually at line 15, make use of polymorphism? (Choose three.)

- A. 

```
public void add(C c) { c.getValue(); }
```
- B. 

```
public void add(B b) { b.getValue(); }
```
- C. 

```
public void add(A a) { a.getValue(); }
```

**Soal Code C**

```
D. public void add(A a, B b) { a.getValue();
}
E. public void add(C c1, C c2) {
c1.getValue(); }
```

**Q: 138 Given:**

```
20. public class CreditCard {
21.
22. private String cardID;
23. private Integer limit;
24. public String ownerName;
25.
26. public void setCardInformation(String cardID,
27. String ownerName,
28. Integer limit) {
29. this.cardID = cardID;
30. this.ownerName = ownerName;
31. this.limit = limit;
32. }
33. }
```

Which statement is true?

- A. The class is fully encapsulated.
- B. The code demonstrates polymorphism.
- C. The ownerName variable breaks encapsulation.
- D. The cardID and limit variables break polymorphism.
- E. The setCardInformation method breaks encapsulation.

**Q: 139**

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 here	Animal	is-a	Dog
Forest	Place here	Place here	has-a	Side
Rectangle	Place here	Place here		Tail
Place here	Place here	Programming Book		Square
				Tree
				Book
				Java Book
				Pen
		Done		

**Q: 140 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

**Q: 141 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.

**Q: 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 A only if the subclass is in the same package as X.

**Q: 143**

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	Modifiers
public class Single {	
private static Single instance;	final
public static Single getInstance() {	protected
if (instance == null) instance = create();	private
return instance;	abstract
}	static
private Single() { }	
protected Single create() { return new Single(); }	
}	
class SingleSub extends Single {	
}	
	Done

**Q: 144 What two must the programmer do to correct the compilation errors? (Choose two.)**

**Soal Code C**

```

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. }

```

- 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;

**Q: 145 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: 146 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.

**Q: 147**

```

Given:
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.

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

**Q: 148 What is the output if the main() method is run?**

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. }

```

- 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.

**Q: 149 Given:**

```

11. Runnable r = new Runnable() {
12.     public void run() {
13.         System.out.print("Cat");
14.     }
15. };
16. Thread t = new Thread(r) {

```

**Soal Code C**

```
17. public void run() {
18. System.out.print("Dog");
19. }
20. };
21. t.start();
```

What is the result?

- A. Cat
- B. Dog
- C. Compilation fails.
- D. The code runs with no output.
- E. An exception is thrown at runtime.

**Q: 150 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 }

**Q: 151**

Place a Class on each method that is declared in the class.

Method Name	Class
run()	java.lang.Object
wait()	java.lang.Thread
notify()	
sleep()	
start()	
join()	

Done

**Q: 152 Given:**

foo and bar are public references available to many other threads. foo refers to a Thread and bar is an Object. The thread foo is currently executing bar.wait().

From another thread, what provides the most reliable way to ensure that foo will stop executing wait()?

- A. foo.notify();
- B. bar.notify();
- C. foo.notifyAll();
- D. Thread.notify();
- E. bar.notifyAll();
- F. Object.notify();

**Q: 153 Which two statements are true if this class is compiled and run? (Choose two.)**

```
1. import java.util.*;
2.
3. public class NameList {
4.     private List names = new ArrayList();
5.     public synchronized void add(String
name) { names.add(name); }
6.     public synchronized void printAll() {
7.         for (int i = 0; i < names.size();
i++) {
8.             System.out.print(names.get(i) + "
");
9.         }
10.    }
11.    public static void main(String[] args)
{
12.        final NameList sl = new NameList();
13.        for (int i = 0; i < 2; i++) {
14.            new Thread() {
15.                public void run() {
16.                    sl.add("A");
17.                    sl.add("B");
18.                    sl.add("C");
19.                    sl.printAll();
20.                }
21.            }.start();
22.        }
23.    }
24. }
```

- A. An exception may be thrown at runtime.
- B. The code may run with no output, without exiting.
- C. The code may run with no output, exiting normally.
- D. The code may run with output "A B A B C C", then exit
- E. The code may run with output "A B C A B C A B C", then exit.
- F. The code may run with output "A A A B C A B C C", then exit.
- G. The code may run with output "A B C A A B C A B C", then exit.

**Q: 154 Given:**

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

### Soal Code C

```
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.

#### Q: 155 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.

#### Q: 156 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.

#### Q: 157 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) {
22.         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: 158 Given:

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

What is the result if a NullPointerException occurs on line 34?

- A. c
- B. a
- C. ab
- D. ac
- E. bc
- F. abc

#### Q: 159 Given:

```
11. public static Collection get() {
12.     Collection sorted = new LinkedList();
13.     sorted.add("B");
14.     sorted.add("C");
15.     sorted.add("A");
16.     return sorted;
17. }
18. public static void main(String[] args) {
19.     for (Object obj: get()) {
20.         System.out.print(obj + ", ");
21.     }
22. }
```

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: 160 Given:

```
23. int z = 5;
24.
25. public void stuff1(int x) {
26.     assert (x > 0);
27.     switch(x) {
28.     case 2: x = 3;
29.     default: assert false; } }
30.
31. private void stuff2(int y) { assert (y < 0); }
32.
33. private void stuff3() { assert (stuff4()); }
34.
35. private boolean stuff4() { z = 6; return false; }
}
```

Which statement is true?

- A. All of the assert statements are used appropriately.
- B. Only the assert statement on line 31 is used appropriately.
- C. The assert statements on lines 29 and 31 are used appropriately.
- D. The assert statements on lines 26 and 29 are used appropriately.
- E. The assert statements on lines 29 and 33 are used appropriately.
- F. The assert statements on lines 29, 31, and 33 are used appropriately.

### Soal Code C

- G. The assert statements on lines 26, 29, and 31 are used appropriately.

#### Q: 161 Given:

```
11. public class Test {
12.     public static void main(String [] args) {
13.         int x = 5;
14.         boolean b1 = true;
15.         boolean b2 = false;
16.
17.         if ((x == 4) && !b2 )
18.             System.out.print("1 ");
19.         System.out.print("2 ");
20.         if ((b2 = true) && b1 )
21.             System.out.print("3 ");
22.     }
23. }
```

What is the result?

- A. 2
- B. 3
- C. 12
- D. 23
- E. 123
- F. Compilation fails.
- G. An exception is thrown at runtime.

#### Q: 162 Given:

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

What is the result?

```
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. }
```

- A. Compilation fails.
- B. ClassC is displayed.
- C. The code runs with no output.
- D. An exception is thrown at runtime.

#### Q: 163 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?

```
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. }
```

- 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.

#### Q: 164 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");
}
```

#### Q: 165 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.

### Soal Code C

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

**Q: 166 Which four statements are true?** (Choose four.)

- A. Has-a relationships should never be encapsulated.
- B. Has-a relationships should be implemented using inheritance.
- C. Has-a relationships can be implemented using instance variables.
- D. Is-a relationships can be implemented using the extends keyword.
- E. Is-a relationships can be implemented using the implements keyword.
- F. The relationship between Movie and Actress is an example of an is-a relationship.
- G. An array or a collection can be used to implement a one-to-many has-a relationship.

**Q: 167 Given:**

```
20. public class CreditCard {
21.
22. private String cardID;
23. private Integer limit;
24. public String ownerName;
25.
26. public void setCardInformation(String cardID,
27. String ownerName,
28. Integer limit) {
29. this.cardID = cardID;
30. this.ownerName = ownerName;
31. this.limit = limit;
32. }
33. }
```

Which statement is true?

- A. The class is fully encapsulated.
- B. The code demonstrates polymorphism.
- C. The ownerName variable breaks encapsulation.
- D. The cardID and limit variables break polymorphism.
- E. The setCardInformation method breaks encapsulation.

**Q: 168 Given:**

```
10. abstract class A {
11. abstract void a1();
12. void a2() { }
13. }
14. class B extends A {
15. void a1() { }
16. void a2() { }
17. }
18. class C extends B { void c1() { } }
```

and:

```
A x = new B(); C y = new C(); A z = new C();
```

What are four valid examples of polymorphic method calls? (Choose four.)

- A. x.a2();
- B. z.a2();
- C. z.c1();
- D. z.a1();
- E. y.c1();
- F. x.a1();

**Q: 169 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

**Q: 170 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; }

**Q: 171**

### Soal Code C

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: 172 Given:**

```
10. public class SuperCalc {
11.     protected static int multiply(int a, int b) {
12.         return a * b;
13.     }
14. }
```

and:

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

and:

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

What is the result?

- A. 12
- B. 4
- C. The code runs with no output.
- D. An exception is thrown at runtime.
- E. Compilation fails because of an error in line 21.
- F. Compilation fails because of an error in line 22.
- G. Compilation fails because of an error in line 31.

**Q: 173 Given:**

```
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?

Given:

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

And:

```
30. class PepperoniPizza extends Pizza {
31.     public void addTopping(String topping) {
32.         System.out.println("Cannot add
33.         Toppings");
34.     }
35.     public void removeTopping(String
36.         topping) {
37.         System.out.println("Cannot remove
38.         Pepperoni");
39.     }
```

And:

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

- A. Compilation fails.
- B. Cannot add Toppings
- C. The code runs with no output.
- D. A NullPointerException is thrown in Line 4.

**Q: 174 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 A only if the subclass is in the same package as X.

**Q: 175 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);`

**Soal Code C**

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

**Q: 176 Given:**

```
1. public class MyLogger {
2. private StringBuilder logger =
   new StringBuuilder();
3. public void log(String message, String user) {
4. logger.append(message);
5. logger.append(user);
6. }
7. }
```

The programmer must guarantee that a single MyLogger object works properly for a multi-threaded system. How must this code be changed to be thread-safe?

- A. synchronize the log method
- B. replace StringBuilder with StringBuffer
- C. replace StringBuilder with just a String object and use the string concatenation (+) within the log method
- D. No change is necessary, the current MyLogger code is already thread-safe.

**Q: 177**

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

```
reader =   ("in" );
writer =    ("out" );
```

**Constructors**

- 
- 
- 
- 
- 
- 

**Q: 178 Which code, inserted at line 14, will allow this class to correctly serialize and deserialize?**

```
1. import java.io.*;
2. public class Foo implements Serializable
{
3.     public int x, y;
4.     public Foo( int x, int y ) { this.x =
x; this.y = y; }
5.
6.     private void writeObject(
ObjectOutputStream s )
7.         throws IOException {
8.         s.writeInt(x); s.writeInt(y) ;
9.     }
10.
11.    private void readObject(
ObjectInputStream s )
12.        throws IOException,
ClassNotFoundException {
13.
14.        // insert code here
15.
16.    }
17. }
```

- A.
- B.
- C.

```
D. 
```

**Q: 179 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: 180**

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) {

for ( String dir : directories ) {

}


}
```

**Code Fragments**

- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
-