1Z0-808 Mock Exam

ExamId: 100
Items: 56
Dificulty: HARD

August 5, 2025

(1) (questionId: 100127) Which of the following ‘main‘ method signatures will cause a ‘java.lang.NoSuchMethodError: main‘ exception at runtime, assuming the class is compiled successfully? (Choose all that apply) Choose all the correct answer.

- 0) ‘public static void main(String... args)‘
- 1) ‘public void main(String[] args)‘
- 2) ‘public static void Main(String[] args)‘
- 3) ‘public static void main(String args)‘
- 4) ‘public static int main(String[] args)‘

(2) (questionId: 100926) Examine this code carefully. What is the result?

```
public class Test {
    public static void main(String[] args) {
        Integer i = 128;
        Integer j = 128;
        int k = 128;

        if (i == j) {
            System.out.print("A");
        }
        if (i == k) {
            System.out.print("B");
        }
    }
}
```

Choose the most correct answer.

- 0) A
- 1) B
- 2) AB
- 3) No output is produced.

(3) (questionId: 102427) What is the final value of `sum`?

```
long[][] arr = new long[2][2];
arr[0] = new long[]{1, 2};
arr[1] = arr[0];
arr[0][0] = 5;
long sum = arr[0][0] + arr[1][0];
```

Choose the most correct answer.

- 0) 6
- 1) 7
- 2) 10

- 3) Compilation fails.

(4) (questionId: 101026) What will be printed after this code executes?

```
String[] data = {"a", "b", "c"};
int x = 0;
for(;;){
    try {
        System.out.print(data[x++]);
    } catch (ArrayIndexOutOfBoundsException e) {
        break;
    }
}
```

Choose the most correct answer.

- 0) abc
- 1) ab
- 2) a
- 3) An infinite loop occurs.

(5) (questionId: 101129) Given the following code, which statements are true? (Choose all that apply)

```
public class Test {
    public static void main(String... args) {
        String result = "";
        loop:
        for (int i=0; i<4; i++) {
            if (i % 2 == 0) {
                continue;
            }
            switch(i) {
                case 1: result += "A"; break;
                case 3: result += "B"; break loop;
                case 5: result += "C";
            }
            result += "D";
        }
        System.out.println(result);
    }
}
```

Choose all the correct answer.

- 0) The ‘continue‘ statement is executed when ‘i‘ is 0 and 2.
- 1) The code enters the ‘switch‘ statement when ‘i‘ is 1 and 3.
- 2) The string ‘"D"‘ is appended to ‘result‘ exactly once.
- 3) The ‘break loop;‘ statement is executed.

- 4) The final output is ‘ABD‘.
- 5) The final output is ‘AB‘.

(6) (questionId: 102223) What is the result of attempting to access ‘MyDevice.NAME‘ in another class?

```
interface Device {
    String NAME = "Device";
}
interface Gadget {
    String NAME = "Gadget";
}
class MyDevice implements Device, Gadget {
    // Some code
}
// In another class:
// System.out.println(MyDevice.NAME);
```

Choose the most correct answer.

- 0) It prints ”Device”.
- 1) It prints ”Gadget”.
- 2) It results in a compile-time error due to an ambiguous field.
- 3) It prints ‘null‘.

(7) (questionId: 100029) Which of these are valid command line argument arrays in a main method signature? (Choose all that apply) Choose all the correct answer.

- 0) `String args[]`
- 1) `String... args`
- 2) `String[] myArgs`
- 3) `String[] _args`
- 4) `String..._args`

(8) (questionId: 101826) Select all lines of code after which at least one ‘Gadget‘ object becomes eligible for garbage collection.

```
class Gadget {}
public class GadgetFactory {
    static Gadget staticGadget = new Gadget(); // Line 1
    Gadget instanceGadget = new Gadget();      // Line 2

    public static void main(String[] args) {
        GadgetFactory gf = new GadgetFactory(); // Line 3
        Gadget g1 = new Gadget();               // Line 4
        gf.build(g1);
        g1 = null;                              // Line 5
        gf = null;                              // Line 6
    }

    void build(Gadget g) {
        Gadget g2 = new Gadget();               // Line 7
    } // End of build method is effectively Line 8
}
```

Choose all the correct answer.

- 0) Line 5
- 1) Line 6
- 2) Line 8
- 3) The line after the ‘main‘ method completes.
- 4) Line 3

(9) (questionId: 102722) What is the result of this code?

```
Comparator<Integer> c = (i1, i2) -> i1 - i2;
List<Integer> list = Arrays.asList(Integer.MAX_VALUE, Integer.MIN_VALUE);
Collections.sort(list, c);
System.out.println(list);
```

Choose the most correct answer.

- 0) ‘[-2147483648, 2147483647]‘
- 1) ‘[2147483647, -2147483648]‘
- 2) An ‘ArithmeticException‘ is thrown.
- 3) The list remains unchanged.

(10) (questionId: 100723) Consider the following class. What is the outcome?

```
public class Test {
    static {
        i = 20; // Forward reference is ok in assignment
    }
    static int i = 10;

    public static void main(String[] args) {
        System.out.println(i);
    }
}
```

Choose the most correct answer.

- 0) 20
- 1) 10
- 2) Compilation fails due to illegal forward reference.
- 3) 0

(11) (questionId: 102126) What is the result of attempting to compile this code snippet?

```
import java.util.*;

public class GenericsTest {
    public static void main(String[] args) {
        List<String> stringList = new ArrayList<>();
        if (stringList instanceof List<Integer>) {
            System.out.println("It’s a list of Integers");
        }
    }
}
```

Choose the most correct answer.

- 0) The code compiles and runs, but the ‘if‘ block is never executed.
- 1) The code compiles and throws a ‘ClassCastException‘ at runtime.
- 2) A compile-time error occurs.
- 3) The code compiles and runs, and the ‘if‘ block is executed due to type erasure.

(12) (questionId: 100421) What is the result of attempting to compile the following code snippet?

```
int i = 10;
byte b = i;
```

Choose the most correct answer.

- 0) It compiles successfully because the value of ‘i‘ (10) is within the range of a ‘byte‘.
- 1) It fails to compile because ‘i‘ is an ‘int‘ variable, and assigning it to a ‘byte‘ requires an explicit cast.
- 2) It compiles, but will throw a runtime exception if ‘i‘ were greater than 127.
- 3) It compiles because the compiler can determine the constant value of ‘i‘ at compile time.

(13) (questionId: 101227) Which of the following are true about enums in Java? (Choose all that apply) Choose all the correct answer.

- 0) An enum can be a generic type, e.g., ‘public enum MyEnum¡T¿ ... ‘
- 1) Enum constants are implicitly ‘public‘, ‘static‘, and ‘final‘.
- 2) An enum can contain a ‘main‘ method and can be executed as a standalone program.
- 3) An enum type cannot be a subtype of another enum.

(14) (questionId: 101622) What is the output of this program?

```
public class ForwardReference {
    {
        System.out.print(value + " ");
    }
    private int value = 1;
    {
        System.out.print(value + " ");
    }

    public ForwardReference() {
        System.out.print(value);
    }

    public static void main(String... args) {
        new ForwardReference();
    }
}
```

Choose the most correct answer.

- 0) 1 1 1
- 1) 0 1 1
- 2) 0 0 1
- 3) The code fails to compile.

(15) (questionId: 102022) What is the result?

```
class SuperClass {
    static String ID = "Super";
    void printID() { System.out.println(ID); }
}

class SubClass extends SuperClass {
    static String ID = "Sub";
    void printID() { System.out.println(ID); }
}

public class TestHiding {
    public static void main(String[] args) {
        SuperClass sup = new SubClass();
        System.out.println(sup.ID);
        sup.printID();
    }
}
```

Choose the most correct answer.

- 0) Super
- 1) Sub

- 2) Super
- 3) Sub
- 4) Compilation fails.

(16) (questionId: 103122) An exception is thrown from a ‘try-with-resources‘ block, another from the resource’s ‘close()‘ method, and a third from the ‘finally‘ block. Which exception is ultimately propagated to the caller? Choose the most correct answer.

- 0) The exception from the ‘try‘ block.
- 1) The exception from the ‘close()‘ method.
- 2) The exception from the ‘finally‘ block.
- 3) A new wrapper exception containing all three.

(17) (questionId: 100424) What value is stored in the variable ‘result‘ after this code is executed?

```
long result = 2_147_483_647 + 1;
```

Choose the most correct answer.

- 0) ‘2147483648‘
- 1) ‘-2147483648‘
- 2) The code fails to compile.
- 3) ‘21474836471‘

(18) (questionId: 103452) What is the output of the following code, which uses a statically imported nested class?

```
// File: Encloser.java
public class Encloser {
    public static class Nested {
        public void hi() { System.out.println("Hi"); }
    }
}

// File: Main.java
import static Encloser.Nested;

public class Main {
    public static void main(String[] args) {
        Nested n = new Nested();
        n.hi();
    }
}
```

Choose the most correct answer.

- 0) ‘Hi‘

- 1) The code fails to compile because you cannot statically import a class.
- 2) The code fails to compile because ‘Nested‘ must be instantiated via ‘Encloser.Nested‘.
- 3) The code fails to compile for a different reason.

(19) (questionId: 100722) What will the following code print?

```
public class ScopePuzzle {
    int x = 5;

    public static void main(String[] args) {
        ScopePuzzle p = new ScopePuzzle();
        p.go();
    }

    void go() {
        int x;
        go2();
        // System.out.println(x); // Line X
    }

    void go2() {
        x = 10;
    }
}
```

Choose the most correct answer.

- 0) If Line X is uncommented, the code will print 10.
- 1) If Line X is uncommented, the code will print 5.
- 2) If Line X is uncommented, the code will fail to compile.
- 3) The code as is will compile and run without error.

(20) (questionId: 102320) What is the output of the following code?

```
public class Finalizer {
    private final int value;
    public Finalizer(int v) {
        this.value = v;
    }
    public int getValue() {
        return this.value;
    }
    public static void main(String[] args) {
        final Finalizer f = new Finalizer(20);
        // Line X
        System.out.println(f.getValue());
    }
    public void modify(Finalizer fin) {
        fin = new Finalizer(30);
    }
}
```

What would happen if ‘modify(f);‘ was inserted at ’Line X’? Choose the most correct answer.

- 0) The code would fail to compile because ‘f‘ is final.
- 1) The code would print 30.
- 2) The code would print 20.
- 3) The code would throw a runtime exception.

(21) (questionId: 100924) What are the final values of ‘x‘ and ‘y‘ after this code snippet runs?

```
int x = 10;
int y = 20;
if (++x <= 10 && --y > 15) {
    x++;
    y++;
}
```

Choose the most correct answer.

- 0) ‘x‘ is 11, ‘y‘ is 20
- 1) ‘x‘ is 11, ‘y‘ is 19
- 2) ‘x‘ is 12, ‘y‘ is 20
- 3) ‘x‘ is 10, ‘y‘ is 20

(22) (questionId: 101627) What is the result of compiling this class?

```
public class FinalChallenge {
    private final int value;

    public FinalChallenge() {
        this(10);
        // value = 20; // Line A
    }

    public FinalChallenge(int value) {
        this.value = value;
    }
}
```

Choose the most correct answer.

- 0) The code compiles successfully as is.

- 1) The code fails to compile because a final field is assigned in one constructor but not the other.
- 2) If Line A is uncommented, the code will fail to compile.
- 3) The code fails to compile because a final field cannot be assigned in a constructor that uses ‘this()‘.

(23) (questionId: 101729) Given ‘public class Test static int x = 1; int y = 2; ‘, which of the following lines of code are valid if placed inside the ‘main‘ method of another class? (Choose all that apply) Choose all the correct answer.

- 0) ‘System.out.println(Test.x);‘
- 1) ‘System.out.println(Test.y);‘
- 2) ‘Test t = new Test(); System.out.println(t.x);‘
- 3) ‘Test t = new Test(); System.out.println(t.y);‘
- 4) ‘Test t = null; System.out.println(t.x);‘
- 5) ‘Test t = null; System.out.println(t.y);‘

(24) (questionId: 100524) What is the final value of ‘s‘?

```
short s = 32767;
s++;
```

Choose the most correct answer.

- 0) ‘32768‘
- 1) ‘-32768‘
- 2) ‘0‘
- 3) The code does not compile.

(25) (questionId: 100827) Which statement best describes the evaluation of the following expression?

```
int a = 1, b = 2, c = 3, d = 4;
int result = a + b * c / d > a ? b + c : d - a;
```

Choose the most correct answer.

- 0) The expression evaluates to 5.
- 1) The expression evaluates to 3.
- 2) The multiplication ‘b*c‘ is performed first.
- 3) The ternary operator ‘? :‘ has higher precedence than ‘¿‘.

(26) (questionId: 101924) Given the code:

```
// In package company.parts
package company.parts;
public class Engine {
    // package-private constructor
    Engine() {}
}

// In package company.parts
package company.parts;
public class PartsFactory {
    public static Engine getEngine() {
        return new Engine();
    }
}

// In package company.vehicles
package company.vehicles;
import company.parts.*;
public class Car {
    public static void main(String[] args) {
        Engine e = PartsFactory.getEngine(); // Line X
        System.out.println("Engine acquired");
    }
}
```

What is the result? Choose the most correct answer.

- 0) Compilation fails at Line X because ‘Engine‘’s constructor is not visible.
- 1) Compilation fails at Line X because the ‘Engine‘ class is not visible.
- 2) Compilation succeeds, and ”Engine acquired” is printed.
- 3) Compilation fails because ‘PartsFactory.getEngine()‘ returns a type whose constructor is not public.

(27) (questionId: 101528) Which statements are true regarding the initialization of a new object? (Choose all that apply) Choose all the correct answer.

- 0) The constructor body is executed before instance initializers.
- 1) If present, a call to another constructor using ‘this()‘ must be the very first statement in a constructor.
- 2) Static variables are initialized after the constructor completes.
- 3) Instance variables are assigned their default values (e.g., 0, false, null) before any instance initializers or constructors are run.
- 4) Instance initializers are executed in the order they appear in the source code.
- 5) It is valid for a class to have multiple instance initializer blocks.

(28) (questionId: 102520) What is the result of executing the following code?

```
import java.util.List;
import java.util.ArrayList;
```

```
public class Test {
    public static void main(String[] args) {
        List<Integer> list = new ArrayList<>();
        list.add(1);
        list.add(2);
        list.add(3);
        list.remove(2);
        System.out.println(list);
    }
}
```

Choose the most correct answer.

- 0) [1, 2]
- 1) [1, 3]
- 2) [2, 3]
- 3) An ‘IndexOutOfBoundsException‘ is thrown.

(29) (questionId: 101425) Which line of code, when inserted at ‘// INSERT‘, will result in both ‘boolean‘ variables being ‘true‘?

```
StringBuilder sb1 = new StringBuilder("A");
StringBuilder sb2 = new StringBuilder("A");
String s1 = new String("A");

// INSERT

boolean b1 = sb1.toString().equals(s1);
boolean b2 = sb1 == sb2;
```

Choose the most correct answer.

- 0) ‘sb2 = sb1;‘
- 1) ‘sb1 = new StringBuilder(s1);‘
- 2) ‘s1 = sb1.toString(); sb2 = sb1;‘
- 3) It’s impossible to make both ‘true‘.

(30) (questionId: 102928) What is the final output of this program?

```
public class Test {
    public static void main(String[] args) {
        try {
            System.out.print("A");
            danger();
        } catch (Exception e) {
            System.out.print("B");
        } finally {
            System.out.print("C");
        }
    }
    static void danger() {
        try {
            throw new Error();
        } finally {
            System.out.print("D");
        }
    }
}
```

Choose the most correct answer.

- 0) ‘ADBC‘
- 1) ‘ADC‘
- 2) ‘AD‘ followed by an ‘Error‘ being thrown.
- 3) ‘A‘ followed by an ‘Error‘ being thrown.

(31) (questionId: 100125) Consider the following code:

```
package com.test;
public class Runner {
    public static void main(String[] args) {
        System.out.println("OK");
    }
}
```

After compiling with ‘javac -d . com/test/Runner.java‘, you are in the ‘com/test‘ directory. You execute ‘java Runner‘. What is the result? Choose the most correct answer.

- 0) It prints ”OK”.
- 1) A ‘ClassNotFoundException‘ is thrown.
- 2) A ‘NoClassDefFoundError‘ is thrown.
- 3) A ‘SecurityException‘ is thrown.

(32) (questionId: 100227) Which of the following statements about ‘import‘ declarations are true? (Choose all that apply) Choose all the correct answer.

- 0) ‘import‘ statements are required to use any class outside the current package.
- 1) A static import can import all static members of a class using a wildcard (‘*‘).
- 2) Importing a package, such as ‘java.util.*‘, also imports its subpackages, like ‘java.util.concurrent‘.
- 3) Importing a class with the same simple name from two different packages requires one of them to be referred to by its fully qualified name.

- 4) ‘import‘ statements increase the size of the final ‘.class‘ file.

(33) (questionId: 100327) What is the result of compiling and running this code?

```
public class TrickyScope {
    public static void main(String[] args) {
        int i = 0;
        if (true) {
            // The following comment looks like it closes the block
            /*
                System.out.println("Inside comment");
            }
            */
            i = 1;
        }
        System.out.println(i);
    }
}
```

Choose the most correct answer.

- 0) It fails to compile due to a syntax error with braces.
- 1) It compiles and prints ‘0‘.
- 2) It compiles and prints ‘1‘.
- 3) It compiles but throws a runtime exception.

(34) (questionId: 103229) Which of the following functional interface declarations will compile successfully? (Choose all that apply) Choose all the correct answer.

- 0) ‘@FunctionalInterface interface A int m(); default int n() return 0; ‘
- 1) ‘@FunctionalInterface interface B extends A ‘
- 2) ‘@FunctionalInterface interface C ¡T¿ T m(T t); ‘
- 3) ‘@FunctionalInterface interface D extends java.util.Comparator ‘
- 4) ‘@FunctionalInterface interface E void m(); String toString(); ‘

(35) (questionId: 101529) You have an encapsulated ‘MutableDate‘ class. Which of the following getter method implementations for a ‘Person‘ class would risk breaking the encapsulation of the ‘Person‘ object’s state? (Choose all that apply)

```
// Assume MutableDate is a class like java.util.Date
// with public methods to change its state.
class MutableDate { /* ... setters ... */ }

class Person {
    private String name;
    private MutableDate birthDate;

    public Person(String name, MutableDate birthDate) {
        this.name = name;
        this.birthDate = birthDate;
    }

    // ... getters ...
}
```

Choose all the correct answer.

- 0) ‘public MutableDate getBirthDate() return this.birthDate; ‘
- 1) ‘public String getName() return this.name; ‘
- 2) ‘public MutableDate getBirthDate() return new MutableDate(this.birthDate.getTime()); ‘
- 3) ‘public MutableDate getBirthDate() return (MutableDate) this.birthDate.clone(); ‘ (Assume ‘clone()‘ is implemented correctly for a deep copy).
- 4) ‘public void printBirthDate() System.out.println(this.birthDate); ‘

(36) (questionId: 100823) What is the result of this code snippet?

```
int mask = 0x000F;
int value = 0x2222;
System.out.println(value & mask);
```

Choose the most correct answer.

- 0) 15
- 1) 2
- 2) 0
- 3) 2222

(37) (questionId: 101327) Which statements are true about string concatenation using the ‘+‘ operator in a loop? (Choose all that apply)

```
String result = "";
for (int i=0; i<100; i++) {
    result += i; // Line 3
}
```

Choose all the correct answer.

- 0) A new ‘String‘ object is created in each iteration of the loop.
- 1) The compiler automatically replaces this code with ‘StringBuilder‘ for efficiency.
- 2) This is the most memory-efficient way to build a string.
- 3) After the loop, the original ‘result‘ object (the empty string) has been modified to contain the final value.

(38) (questionId: 101828) Analyze the following code. At Point Y, how many ‘java.lang.String‘ objects are eligible for GC, assuming no string pooling optimizations for literals?

```
public class StringGC {
    public static void main(String[] args) {
        String s1 = "one";
        String s2 = new String("two");
        String s3 = "three";
        s3 = s1;
        s1 = s2;
        s2 = null;

        // What about the object referred to by s1 originally ("one")?
        // What about the object referred to by s2 originally ("two")?
        // What about the object referred to by s3 originally ("three")?
        // Point Y
    }
}
```

Choose the most correct answer.

- 0) 0
- 1) 1
- 2) 2
- 3) 3

(39) (questionId: 103022) What is the result of attempting to compile and run the following code?

```
public class StaticFail {
    static {
        if (true) {
            throw new RuntimeException("Initialization failed");
        }
    }

    public static void main(String[] args) {
        System.out.println("Hello");
    }
}
```

Choose the most correct answer.

- 0) The code compiles and prints ‘Hello‘.
- 1) The code does not compile.
- 2) The code compiles, but throws a ‘RuntimeException‘ when run.
- 3) The code compiles, but throws an ‘ExceptionInInitializerError‘ when run.
- 4) The code compiles, but throws a ‘NoClassDefFoundError‘ when run.

(40) (questionId: 101325) What is the output of the following code?

```
String text = "a.b.c";
String[] parts = text.split(".");
System.out.println(parts.length);
```

Choose the most correct answer.

- 0) 0
- 1) 1
- 2) 3
- 3) An exception is thrown at runtime.

(41) (questionId: 103653) What is the output of this code which passes and returns references?

```
class Num { public int val; }

public class ReturnTest {
    public static void main(String[] args) {
        Num a = new Num(); a.val = 1;
        Num b = new Num(); b.val = 2;
        b = process(a, b);
        System.out.println(a.val + "," + b.val);
    }

    public static Num process(Num x, Num y) {
        x.val = y.val;
        y = new Num();
        y.val = 3;
        return y;
    }
}
```

Choose the most correct answer.

- 0) ‘1,2‘
- 1) ‘2,3‘
- 2) ‘2,2‘
- 3) ‘1,3‘

(42) (questionId: 103357) Which of the following lines of code, if executed independently, will result in a runtime exception? (Choose all that apply)

```
// Assume all necessary imports from java.time and java.time.temporal
```

Choose all the correct answer.

- 0) ‘LocalDate.of(2025, 13, 1);‘
- 1) ‘Duration.between(LocalDate.now(), LocalDateTime.now());‘

- 2) ‘Period.of(1, 1, 1).plus(Duration.ofHours(1));‘
- 3) ‘LocalTime.now().truncatedTo(ChronoUnit.DAYS);‘
- 4) ‘Period.ofMonths(12).normalized();‘

(43) (questionId: 100021) Consider the following directory structure and files:

```
/project
    /src
        /com
            /example
                MyClass.java
    /bin
```

The file `MyClass.java` contains:

```
package com.example;

public class MyClass {
    public static void main(String[] args) {
        System.out.println("Running MyClass");
    }
}
```

You are currently in the `/project` directory. Which sequence of commands will successfully compile and run `MyClass`? Choose the most correct answer.

- 0)

  ```
  javac src/com/example/MyClass.java
  java -cp src com.example.MyClass
  ```

- 1)

  ```
  javac src/com/example/MyClass.java
  java -cp bin com.example.MyClass
  ```

- 2)

  ```
  javac -d bin src/com/example/MyClass.java
  java -cp bin com.example.MyClass
  ```

- 3)

  ```
  javac -d bin src/com/example/MyClass.java
  java com.example.MyClass
  ```

(44) (questionId: 100525) Which of the following code snippets will compile successfully? (Choose all that apply) Choose all the correct answer.

- 0)

  ```
  short s = 10;
  s = s + 5;
  ```

- 1)

```
char c = 'a';
c += 5;
```

- 2)

```
final byte b1 = 10;
final byte b2 = 20;
byte b3 = b1 + b2;
```

- 3)

```
float f = 1.0f;
double d = f;
```

(45) (questionId: 100623) Which of the following lines will compile without errors? (Choose all that apply) Choose all the correct answer.

- 0) `Integer i = new Integer(null);`
- 1) `Double d = null; double d2 = d;`
- 2) `Byte b = 25;`
- 3) `Short s = new Short((short)10);`
- 4) `long l = new Integer(100);`

(46) (questionId: 101222) Examine the following code. What is the result?

```
public enum Operation {
    PLUS {
        public double apply(double x, double y) { return x + y; }
    },
    MINUS {
        public double apply(double x, double y) { return x - y; }
    };
    public abstract double apply(double x, double y);
}

class Test {
    public static void main(String[] args) {
        System.out.println(Operation.PLUS.apply(5, 3));
    }
}
```

Choose the most correct answer.

- 0) ‘8.0‘
- 1) The code fails to compile because an enum cannot be ‘abstract‘.
- 2) The code fails to compile because ‘apply‘ is not defined for the ‘Operation‘ enum itself.
- 3) The code fails to compile because an enum constant cannot provide a method implementation.

(47) (questionId: 100328) Which of the following code snippets will fail to compile due to issues with comment syntax? (Choose all that apply) Choose all the correct answer.

- 0)

  ```
  int x = 10; //* A special comment */
  ```

- 1)

  ```
  String s = "This contains a comment end: */";
  ```

- 2)

  ```
  /* Is this /* nested comment */ valid? */
  int y = 20;
  ```

- 3)

  ```
  // Another comment \
  int z = 30;
  ```

(48) (questionId: 102622) Which of these lines causes a compilation error?

```
import java.util.*;

class Mammal {}
class Primate extends Mammal {}
class Human extends Primate {}

public class Test {
    public static void main(String[] args) {
        List<? super Primate> primates = new ArrayList<Mammal>(); // Line 1
        primates.add(new Human());                                // Line 2
        primates.add(new Primate());                              // Line 3
        primates.add(new Mammal());                               // Line 4
    }
}
```

Choose the most correct answer.

- 0) Line 1
- 1) Line 2
- 2) Line 3
- 3) Line 4

(49) (questionId: 101025) What is the output of the following code?

```
int[] a = {1, 2, 3};
int[] b = {4, 5, 6};
for (int i : a, j : b) {
    System.out.print(i + j);
}
```

Choose the most correct answer.

- 0) 579
- 1) 142536
- 2) The code does not compile.
- 3) The code throws a runtime exception.

(50) (questionId: 103558) Which of the following method declarations are valid in a concrete (non-abstract) class? (Choose all that apply) Choose all the correct answer.

- 0) ‘private final static void methodA();‘
- 1) ‘protected abstract void methodB();‘
- 2) ‘public final synchronized void methodC(String... s) ‘
- 3) ‘void methodD(final int... x) ‘
- 4) ‘static System.out.println(”I am not a method.”); ‘

(51) (questionId: 101122) What is the result of attempting to compile this code?

```
public class InvalidContinue {
    public static void main(String[] args) {
        myLabel: {
            if (true) {
                continue myLabel;
            }
        }
    }
}
```

Choose the most correct answer.

- 0) It compiles successfully.
- 1) It fails to compile because ‘myLabel‘ is not on a loop.
- 2) It fails to compile because a label cannot be on a simple block.
- 3) It fails to compile because of an unreachable statement.

(52) (questionId: 100625) What is the output of the following code?

```
import java.util.ArrayList;
import java.util.List;

public class Test {
    public static void main(String[] args) {
        List<Integer> list = new ArrayList<>();
        list.add(1);
        list.add(2);
        list.add(3);
        list.remove(new Integer(2));
```

```
            System.out.println(list);
        }
    }
```

Choose the most correct answer.

- 0) `[1, 2]`
- 1) `[1, 3]`
- 2) `[2, 3]`
- 3) An `IndexOutOfBoundsException` occurs.

(53) (questionId: 101725) What is the output of this code? This tests understanding of ‘this‘ within inner classes.

```
public class Outer {
    String name = "Outer";

    class Inner {
        String name = "Inner";
        void printNames() {
            System.out.println(name);
            System.out.println(this.name);
            System.out.println(Outer.this.name);
        }
    }

    public static void main(String[] args) {
        new Outer().new Inner().printNames();
    }
}
```

Choose the most correct answer.

- 0) Inner
- 1) Outer
- 2) Inner
- 3) The code fails to compile.

(54) (questionId: 100223) The classpath is set to ‘-cp dirA:dirB‘. ‘dirA‘ contains ‘com/test/Tool.class‘ version 1. ‘dirB‘ contains ‘com/test/Tool.class‘ version 2. A program uses ‘com.test.Tool‘. Which version of the class will be loaded by the JVM? Choose the most correct answer.

- 0) Version 1 from ‘dirA‘.
- 1) Version 2 from ‘dirB‘.
- 2) A compilation error will occur.
- 3) A runtime error will occur due to the conflict.

(55) (questionId: 101428) Which of these method calls can throw a ‘StringIndexOutOfBoundsException‘? (Choose all that apply)

```
StringBuilder sb = new StringBuilder("abc");
```

Choose all the correct answer.

- 0) ‘sb.delete(1, 4);‘
- 1) ‘sb.insert(4, ”d”);‘
- 2) ‘sb.replace(0, 5, ”x”);‘
- 3) ‘sb.setCharAt(3, ’d’);‘

(56) (questionId: 102824) What is the result of attempting to compile this code?

```
public class Test {
    public static void main(String[] args) {
        throw new String("This is an error");
    }
}
```

Choose the most correct answer.

- 0) It compiles, but throws a ‘ClassCastException‘ at runtime.
- 1) It compiles, but throws a ‘RuntimeException‘ at runtime.
- 2) It compiles, but throws an ‘Error‘ at runtime.
- 3) It does not compile.