

## 1Z0-819 Dumps

### Java SE 11 Developer

<https://www.certleader.com/1Z0-819-dumps.html>



**NEW QUESTION 1**

Given:

```
package b;
public class Person {
    protected Person() {                //line 1
    }
}
```

and

```
package a;
import b.Person;
public class Main {                    //line 2
    public static void main(String[] args) {
        Person person = new Person(); //line 3
    }
}
```

Which two allow a.Main to allocate a new Person? (Choose two.)

- A. In Line 1, change the access modifier to privateprivate Person() {
- B. In Line 1, change the access modifier to publicpublic Person() {
- C. In Line 2, add extends Person to the Main classpublic class Main extends Person {and change Line 3 to create a new Main objectPerson person = new Main();
- D. In Line 2, change the access modifier to protectedprotected class Main {
- E. In Line 1, remove the access modifierPerson() {

**Answer: BC****NEW QUESTION 2**

Assuming the Widget class has a getPrice method, this code does not compile:

```
List widgets = List.of(new Widget("Basic Widget", 19.55), // line 1
                        new Widget("Enhanced Widget", 35.00),
                        new Widget("Luxury Edition Widget", 55.45));
Stream widgetStream = widgets.stream();                    // line 4
widgetStream.filter(a -> a.getPrice() > 20.00)             // line 5
                .forEach(System.out::println);
```

Which two statements, independently, would allow this code to compile? (Choose two.)

- A. Replace line 5 with widgetStream.filter(a -> ((Widget)a).getPrice() > 20.00).
- B. Replace line 1 with List<Widget> widgetStream = widgets.stream();.
- C. Replace line 5 with widgetStream.filter((Widget a) -> a.getPrice() > 20.00).
- D. Replace line 4 with Stream<Widget> widgetStream = widgets.stream();.

**Answer: AD****NEW QUESTION 3**

Given:

```
package a;
public abstract class Animal {
    protected abstract void walk();
}
package b;
public abstract class Human extends Animal {
    // line 1
}
```

Which two lines inserted in line 1 will allow this code to compile? (Choose two.)

- A. protected void walk(){}.
- B. void walk(){}.
- C. abstract void walk();
- D. private void walk(){}.
- E. public abstract void walk();

**Answer: AE****NEW QUESTION 4**

Given:

```
public class Tester {
    private int x;
    private static int y;
    public static void main(String[] args) {
        Tester t1 = new Tester();
        t1.x = 2;
        Tester.y = 3;
        Tester t2 = new Tester();
        t2.x = 4;
        t2.y = 5;
        System.out.println(t1.x+", "+t1.y);
        System.out.println(t2.x+", "+Tester.y);
        System.out.println(t2.x+", "+t1.y);
    }
}
```

What is the result?

- A. 2,34,34,5
- B. 2,34,54,5
- C. 2,54,54,5
- D. 2,34,54,3

**Answer: C**

**Explanation:**



#### NEW QUESTION 5

Which two statements are true about the modular JDK? (Choose two.)

- A. The foundational APIs of the Java SE Platform are found in the java.base module.
- B. An application must be structured as modules in order to run on the modular JDK.
- C. It is possible but undesirable to configure modules' exports from the command line.
- D. APIs are deprecated more aggressively because the JDK has been modularized.

**Answer: AC**

#### NEW QUESTION 6

Given:

```
public static void main(String[] args) {
    final List<String> fruits =
        List.of("Orange", "Apple", "Lemmon", "Raspberry");
    final List<String> types =
        List.of("Juice", "Pie", "Ice", "Tart");
    final var stream =
        IntStream.range(0, Math.min(fruits.size(), types.size()))
            .mapToObj((i) -> fruits.get(i) + " " + types.get(i) );
    stream. forEach(System.out::println);
}
```

What is the result?

- A. Orange Juice
- B. The compilation fails.
- C. Orange Juice Apple Pie Lemmon Ice Raspberry Tart
- D. The program prints nothing.

**Answer: C**

**Explanation:**

```
12 public class Person {
13     public static void main (String[] args) {
14         final List<String> fruits =
15             List.of("Orange", "Apple", "Lemmon", "raspberry");
16         final List<String> types =
17             List.of("Juice", "Pie", "Ice", "Tart");
18         final var stream =
19             IntStream.range(0, Math.min(fruits.size(), types.size()))
20                 .mapToObj ((i) -> fruits.get(i) + " " + types.get(i) );
21         stream. forEach(System.out::println);
22     }
23
24 }
```

**Result**

compiled and executed in 1.227 sec(s)

```
Orange Juice
Apple Pie
Lemmon Ice
raspberry Tart
```

**NEW QUESTION 7**

Given:

```
public class A {
    private boolean checkValue(int val) {
        return true;
    }
}
```

and

```
public class B extends A {
    public int modifyVal(int val) {
        if(checkValue(val)) {
            return val;
        } else {
            return 0;
        }
    }
    public static void Main(String[] args) {
        B b = new B();
        System.out.println(b.modifyVal(10));
    }
}
```

What is the result?

- A. nothing
- B. It fails to compile.
- C. A java.lang.IllegalArgumentException is thrown.
- D. 10

**Answer: B****Explanation:**



```

1- public class A {
2-     private boolean checkValue(int val) {
3-         return true;
4-     }
5- }
6- and
7- public class B extends A {
8-     public int modifyVal(int val) {
9-         if(checkValue(val)) {
10-             return val;
11-         } else {
12-             return 0;
13-         }
14-     }
15-     public static void Main(String[] args) {
16-         B b = new B();
17-         system.out.println(b.modfiyVal (10));
18-     }
19- }

```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: sec(s), Memory: kilobyte(s)

```

/A.java:6: error: class, interface, or enum expected
and
^
1 error

```

#### NEW QUESTION 8

Given:

```

1. {
2.     Iterator iter = List.of(1,2,3).iterator();
3.     while (iter.hasNext()) {
4.         foo(iter.next());
5.     }
6.     Iterator iter2 = List.of(1,2,3).iterator();
7.     while (iter.hasNext()) {
8.         bar(iter2.next());
9.     }
10. }
11. for (Iterator iter = List.of(1,2,3).iterator(); iter.hasNext(); ) {
12.     foo(iter.next());
13. }
14. for (Iterator iter2 = List.of(1,2,3).iterator(); iter.hasNext(); ) {
15.     bar(iter2.next());
16. }

```

Which loop incurs a compile time error?

- A. the loop starting line 11
- B. the loop starting line 7
- C. the loop starting line 14
- D. the loop starting line 3

**Answer: C**

**NEW QUESTION 9**

Examine this excerpt from the declaration of the java.se module:

```
module java.se {  
    ...  
    requires transitive java.sql;  
    ...  
}
```

What does the transitive modifier mean?

- A. Only a module that requires the java.se module is permitted to require the java.sql module.
- B. Any module that requires the java.se module does not need to require the java.sql module.
- C. Any module that attempts to require the java.se module actually requires the java.sql module instead.
- D. Any module that requires the java.sql module does not need to require the java.se module.

**Answer:** A

**NEW QUESTION 10**

Given an application with a main module that has this module-info.java file:

```
module main {  
    exports country;  
    uses country.CountryDetails;  
}
```

Which two are true? (Choose two.)

- A. A module providing an implementation of country.CountryDetails can be compiled and added without recompiling the main module.
- B. A module providing an implementation of country.CountryDetails must have a requires main; directive in its module-info.java file.
- C. An implementation of country.countryDetails can be added to the main module.
- D. To compile without an error, the application must have at least one module in the module source path that provides an implementation of country.CountryDetails.
- E. To run without an error, the application must have at least one module in the module path that provides an implementation of country.CountryDetails.

**Answer:** BD

**NEW QUESTION 10**

Given:

```
public class MethodTest {  
    // line 1  
}
```

Which two method implementations are correct, when inserted independently in line 1? (Choose two.)

- A.  

```
public boolean methodD(int x) {  
    return x > 0;  
}
```
- B.  

```
public String methodB() {  
    System.out.println("methodB");  
}
```
- C.  

```
public char methodE (String msg) {  
    return msg;  
}
```
- D.  

```
public void methodC(int x) {  
    return ++x;  
}
```
- E.  

```
public void methodA() {  
    System.out.println("methodA");  
}
```

- A. Option A
- B. Option B
- C. Option C

- D. Option D  
E. Option E

**Answer:** AE

#### NEW QUESTION 15

Given:

```
1. interface Pastry {  
2.     void getIngredients();  
3. }  
4. abstract class Cookie implements Pastry {}  
5.  
6. class ChocolateCookie implements Cookie {  
7.     public void getIngredients() {}  
8. }  
9. class CoconutChocolateCookie extends ChocolateCookie {  
10.     void getIngredients(int x) {}  
11. }
```

Which is true?

- A. The compilation fails due to an error in line 6.  
B. The compilation succeeds.  
C. The compilation fails due to an error in line 4.  
D. The compilation fails due to an error in line 10.  
E. The compilation fails due to an error in line 7.  
F. The compilation fails due to an error in line 9.  
G. The compilation fails due to an error in line 2.

**Answer:** A

#### NEW QUESTION 16

Given:

```
package A;  
class Test {  
    String name;  
    public Test(String name) {  
        this.name = name;  
    }  
    public String toString() {  
        return name;  
    }  
}
```

and

```
package B;  
import A.Test;  
public class Main {  
    public static void main(String[] args) {  
        Test test = new Test("Student");  
        System.out.println(test);  
    }  
}
```

What is the result?

- A. null  
B. nothing  
C. It fails to compile.  
D. java.lang.IllegalAccessException is thrown.  
E. Student

**Answer:** C

#### NEW QUESTION 19

Given:

```
int arr[][] = {{5,10},{8,12},{9,3}};
long count = Stream.of(arr)
    .flatMapToInt(IntStream::of)
    .map(n -> n + 1)
    .filter(n -> (n % 2 == 0))
    .peek(System.out::print)
    .count();
System.out.println(" " + count);
```

What is the result?

- A. 6910 3
- B. 10126 3
- C. 3
- D. 6104 3

**Answer: D**

**Explanation:**

```
1  import java.util.*;
2  import java.io.*;
3  import java.lang.Thread;
4  import java.util.ArrayList;
5  import java.util.LinkedList;
6  import java.util.List;
7  import java.util.function.Consumer;
8  import java.util.stream.Stream;
9  import java.util.stream.IntStream;
10
11
12 - public class Main {
13
14 -     public static void main(String[] args) {
15         int arr[][] = {{5,10}, {8,12}, {9,3}};
16         long count = Stream.of(arr)
17             .flatMapToInt(IntStream::of)
18             .map(n -> n + 1)
19             .filter(n -> (n % 2 == 0))
20             .peek(System.out::print)
21             .count();
22         System.out.println(" " + count);
23     }
24 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.32 sec(s), Memory: 34220 kilobyte(s)

**6104 3**

**NEW QUESTION 21**

Given:



```
public class Tester {
    static class Person implements /* line 1 */ {
        private String name;
        Person(String name) { this.name = name; }
        /* line 2 */
    }
    public static void main(String[] args) {
        Person[] people = {new Person("Joe"),
                           new Person("Jane"),
                           new Person("John")};
        Arrays.sort(people);
        for(Person person: people) {
            System.out.println(person.name);
        }
    }
}
```

You want the code to produce this output:

John

Joe Jane

Which code fragment should be inserted on line 1 and line 2 to produce the output?

- A. Insert `Comparator<Person>` on line 1. Insert `public int compare(Person p1, Person p2) { return p1.name.compare(p2.name);}` on line 2.
- B. Insert `Comparator<Person>` on line 1. Insert `public int compareTo(Person person) { return person.name.compareTo(this.name);}` on line 2.
- C. Insert `Comparable<Person>` on line 1. Insert `public int compare(Person p1, Person p2) { return p1.name.compare(p2.name);}` on line 2.
- D. Insert `Comparator<Person>` on line 1. Insert `public int compare(Person person) { return person.name.compare(this.name);}` on line 2.

**Answer:** B

#### NEW QUESTION 22

Given:

```
class Mycar {
}
```

and

```
javac C:\workspace4\Mycar.java
```

What is the expected result of javac?

- A. javac fails to compile the class and prints the error message, `C:\workspace4\Mycar.java:1:error: packagejava does not exist`
- B. javac compiles `Mycar.java` without errors or warnings.
- C. javac fails to compile the class and prints the error message, `C:\workspace4\Mycar.java:1:error: expected import java.lang`
- D. javac fails to compile the class and prints the error message, `Error: Could not find or load main class Mycar.class`

**Answer:** B

#### NEW QUESTION 25

Given the formula to calculate a monthly mortgage payment:

$$M = P \frac{r(1+r)^n}{(1+r)^n - 1}$$

and these declarations:

```
double m;           //monthly payment
double r = 0.05/12; //monthly interest rate
int p = 100_000;     //principal
int n = 180;         //number of payments
```

How can you code the formula?

- A. `m = p * (r * Math.pow(1 + r, n) / (Math.pow(1 + r, n) - 1));`
- B. `m = p * ((r * Math.pow(1 + r, n) / (Math.pow(1 + r, n)) - 1));`
- C. `m = p * r * Math.pow(1 + r, n) / Math.pow(1 + r, n) - 1;`
- D. `m = p * (r * Math.pow(1 + r, n) / Math.pow(1 + r, n) - 1);`

**Answer:** A

#### NEW QUESTION 26

Given:

```
public class Main {

    public static void checkConfiguration(String filename) {
        File file = new File(filename);
        if(!file.exists()) {
            throw new Error("Fatal Error: Configuration File, "
                + filename + ", is missing.");
        }
    }

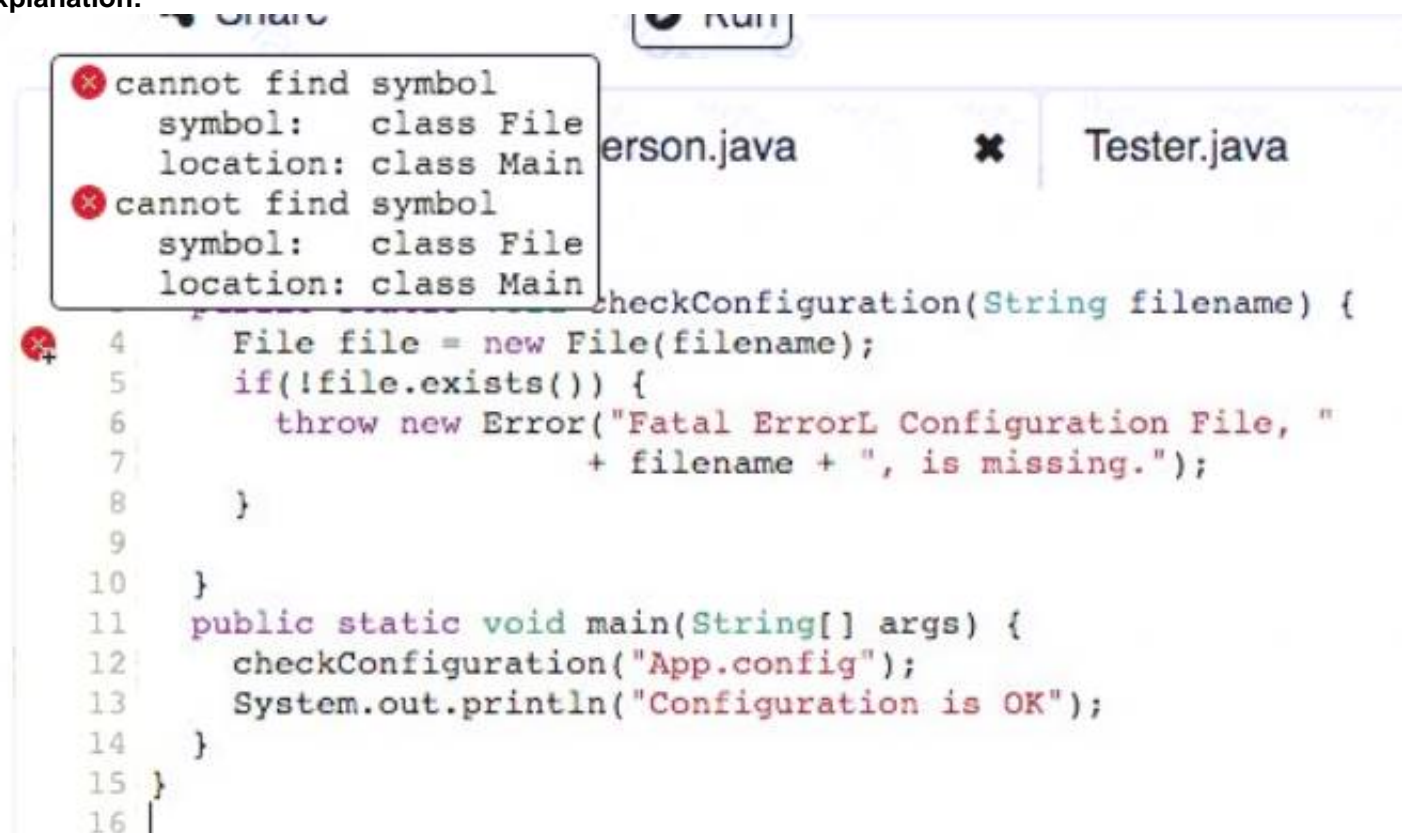
    public static void main(String[] args) {
        checkConfiguration("App.config");
        System.out.println("Configuration is OK");
    }
}
```

If file "App.config" is not found, what is the result?

- A. Configuration is OK
- B. The compilation fails.
- C. Exception in thread "main" java.lang.Error:Fatal Error: Configuration File, App.config, is missing.
- D. nothing

**Answer: B**

**Explanation:**



### NEW QUESTION 30

Given:

```
public interface A {
    abstract void x();
}
```

and

```
public abstract class B /* position 1 */ {
    /* position 2 */
    public void x() { }
    public abstract void z();
}
```

and

```
public class C extends B implements A {
    /* position 3 */
}
```

Which code, when inserted at one or more marked positions, would allow classes B and C to compile?

- A. @Override // position 3  
void x () {} // position 3  
@Override // position 3  
public void z() {} // position 3
- B. @Override // position 2  
public void z() {} // position 3
- C. implements A // position 1  
@Override // position 2
- D. public void z() {} // position 3

**Answer:** B

### NEW QUESTION 33

Given the declaration:

```
@interface Resource {  
    String name();  
    int priority() default 0;  
}
```

Examine this code fragment:

```
/* Loc1 */ class ProcessOrders { ... }
```

Which two annotations may be applied at Loc1 in the code fragment? (Choose two.)

- A. @Resource(priority=100)
- B. @Resource(priority=0)
- C. @Resource(name="Customer1", priority=100)
- D. @Resource(name="Customer1")
- E. @Resource

**Answer:** AB

### NEW QUESTION 37

Which two statements are correct about try blocks? (Choose two.)

- A. A try block can have more than one catch block.
- B. A finally block in a try-with-resources statement executes before the resources declared are closed.
- C. A finally block must be immediately placed after the try or catch blocks.
- D. A try block must have a catch block and a finally block.
- E. catch blocks must be ordered from generic to specific exception types.

**Answer:** AC

### NEW QUESTION 42

Which describes a characteristic of setting up the Java development environment?

- A. Setting up the Java development environment requires that you also install the JRE.
- B. The Java development environment is set up for all operating systems by default.
- C. You set up the Java development environment for a specific operating system when you install the JDK.
- D. Setting up the Java development environment occurs when you install an IDE before the JDK.

**Answer:** D

### NEW QUESTION 43

Given:

```
class MyClass {  
    public static void main(String [] args) {  
        System.out.println(arg[1] + "--" + arg[3] + "--" + arg[0]);  
    }  
}
```

executed using this command: java MyClass My Car is red What is the output of this class?

- A. Car--red--My
- B. My--Car--is
- C. My--is--java
- D. java--Myclass--My
- E. Myclass--Car--red

**Answer:** A

### NEW QUESTION 47

Given:

```
LocalDate d1 = LocalDate.of(1997,2,7);  
DateTimeFormatter dtf = DateTimeFormatter.ofPattern( /*insert code here*/ );  
System.out.println(dtf.format (d1));
```

Which pattern formats the date as Friday 7th of February 1997?

- A. "eeee dd+"th of"+ MMM yyyy"
- B. "eeee dd'th of' MMM yyyy"
- C. "eeee d+"th of"+ MMMM yyyy"
- D. "eeee d'th of' MMMM yyyy"

**Answer:** B



**NEW QUESTION 48**

Given:

```
public class Test {
    public static void doThings() throws GeneralException {
        try {
            throw new RuntimeException("Someting happened");
        } catch (Exception e) {
            throw new SpecificException(e.getMessage());
        }
    }
    public static void main(String args[]) {
        try{
            Test.doThings();
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
class GeneralException /* line 1 */ {
    public GeneralException(String s) { super(s); }
}
class SpecificException /* line 2 */ {
    public SpecificException(String s) { super(s); }
}
```

Which option should you choose to enable the code to print Something happened?

- A. Add extends GeneralException on line 1.Add extends Exception on line 2.
- B. Add extends SpecificException on line 1.Add extends GeneralException on line 2.
- C. Add extends Exception on line 1.Add extends Exception on line 2.
- D. Add extends Exception on line 1.Add extends GeneralException on line 2.

**Answer: D****Explanation:**

```
1  import java.util.*;
2  import java.io.*;
3  import java.lang.Thread;
4  import java.util.ArrayList;
5  import java.util.LinkedList;
6  import java.util.List;
7
8  public class Test {
9
10     public static void doThings() throws GeneralException {
11         try{
12             throw new RuntimeException("Something happened");
13         } catch (Exception e) {
14             throw new SpecificException (e.getMessage());
15         }
16     }
17 }
18
19     public static void main(String args[]) {
20         try{
21             Test.doThings();
22         }catch (Exception e) {
23             System.out.println(e.getMessage());
24         }
25     }
26     class GeneralException extends Exception {
27         public GeneralException(String s) { super(s); }
28     }
29     class SpecificException extends GeneralException {
30         public SpecificException(String s) { super(s);}
31     }
32 }
```

**NEW QUESTION 53**



Given:

```
public static void main(String[] args) {  
    try (Reader reader1 = new FileReader("File1.txt");  
        Reader reader2 = new FileReader("File2.txt");  
        Reader reader3 = new FileReader("File3_txt")) {  
  
    } catch (IOException ex) {  
        Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);  
    }  
    // Line 1  
    System.out.println("Done");  
}
```

When run and all three files exist, what is the state of each reader on Line 1?

- A. All three readers are still open.
- B. All three readers have been closed.
- C. The compilation fails.
- D. Only reader1 has been closed.

**Answer:** C

#### NEW QUESTION 55

Given:

```
class ConSuper {  
    protected ConSuper() {  
        this(2);  
        System.out.print("1");  
    }  
    protected ConSuper(int a) {  
        System.out.print(a);  
    }  
}
```

and

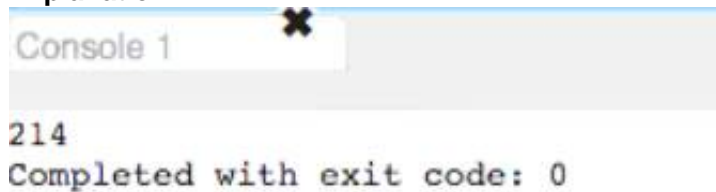
```
public class ConSub extends ConSuper {  
    ConSub() {  
        this(4);  
        System.out.print("3");  
    }  
    ConSub(int a) {  
        System.out.print(a);  
    }  
    public static void main (String[] args) {  
        new ConSub(4);  
    }  
}
```

What is the result?

- A. 2134
- B. 2143
- C. 214
- D. 234

**Answer:** C

**Explanation:**



```
Console 1  
  
214  
Completed with exit code: 0
```

#### NEW QUESTION 57

Consider this method declaration:

```
void setSessionUser(Connection conn, String user) throws SQLException {  
    Statement stmt = conn.createStatement();  
    String sql = <EXPRESSION>;  
    stmt .execute();  
}
```

- A) "SET SESSION AUTHORIZATION " + user  
B) "SET SESSION AUTHORIZATION " + stmt.enquotelIdentifier(user) Is A or B the correct replacement for <EXPRESSION> and why?

- A. A, because it sends exactly the value of user provided by the calling code.  
B. B, because enquoting values provided by the calling code prevents SQL injection.  
C. A and B are functionally equivalent.  
D. A, because it is unnecessary to enclose identifiers in quotes.  
E. B, because all values provided by the calling code should be enquoted.

**Answer:** A

#### NEW QUESTION 58

Given:

```
1. public class Secret {  
2.     String[] names;  
3.     public Secret(String[] names) {  
4.         this.names = names;  
5.     }  
6.     public String[] getNames() {  
7.         return names;  
8.     }  
9. }
```

Which three actions implement Java SE security guidelines? (Choose three.)

- A. Change line 7 to return names.clone();.  
B. Change line 4 to this.names = names.clone();.  
C. Change the getNames() method name to get\$Names().  
D. Change line 6 to public synchronized String[] getNames() {.  
E. Change line 2 to private final String[] names;.  
F. Change line 3 to private Secret(String[] names) {.  
G. Change line 2 to protected volatile String[] names;.

**Answer:** EFG

#### NEW QUESTION 63

Given:

```
import java.io.FileNotFoundException;  
import java.io.IOException;
```

```
public class Tester {  
    public static void main(String[] args) {  
        try {  
            doA();  
        } //line 1  
    }  
    private static void doA() throws IOException, IndexOutOfBoundsException {  
        if (false) {  
            throw new FileNotFoundException();  
        } else {  
            throw new IndexOutOfBoundsException();  
        }  
    }  
}
```

What must be added in line 1 to compile this class?

- A. catch(IOException e) { }  
B. catch(FileNotFoundException | IndexOutOfBoundsException e) { }  
C. catch(FileNotFoundException | IOException e) { }  
D. catch(IndexOutOfBoundsException e) { }catch(FileNotFoundException e) { }  
E. catch(FileNotFoundException e) { }catch(IndexOutOfBoundsException e) { }

**Answer:** A

#### NEW QUESTION 64

Given the code fragment:

```
char[][] arrays = {{'a', 'd'}, {'b', 'e'}, {'c', 'f'}};
for (char[] xx : arrays) {
    for (char yy : xx) {
        System.out.print(yy);
    }
    System.out.print(" ");
}
```

What is the result?

- A. ab cd ef
- B. An `ArrayIndexOutOfBoundsException` is thrown at runtime.
- C. The compilation fails.
- D. abc def
- E. ad be cf

**Answer:** E

#### NEW QUESTION 65

Given:

```
public class FunctionalInterfaceTest {
    public static void main(String[] args) {
        List fruits = Arrays.asList("apple", "orange", "banana");
        Consumer<String> c = System.out::print;
        Consumer<String> output = c.andThen(x -> System.out.println(": " + x.toUpperCase()));
        fruits.forEach(output);
    }
}
```

What is the output?

- A. :APPLE:ORANGE:BANANAappleorangebanana
- B. :APPLE:ORANGE:BANANA
- C. APPLE:apple ORANGE:orange BANANA:banana
- D. appleorangebanana:APPLE:ORANGE:BANANA
- E. apple:APPLE orange:ORANGE banana:BANANA

**Answer:** E

**Explanation:**

```

1  import java.util.*;
2  import java.io.*;
3  import java.lang.Thread;
4  import java.util.ArrayList;
5  import java.util.LinkedList;
6  import java.util.List;
7  import java.util.function.Consumer;
8
9  public class FunctionalInterfaceTest {
10 public static void main (String[] args) {
11     List fruits = Arrays.asList("apple", "orange", "banana");
12     Consumer<String> c = System.out::print;
13     Consumer<String> output = c.andThen(x -> System.out.println(": " + x.toUpperCase()));
14
15     fruits.forEach(output);
16
17 }
18 }

```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4



Interactive

Stdin Inputs

CommandLine Arguments

Execute



Result

CPU Time: 0.26 sec(s), Memory: 32984 kilobyte(s)

```

apple:APPLE
orange:ORANGE
banana:BANANA

```

#### NEW QUESTION 67

A company has an existing sales application using a Java 8 jar file containing packages: com.company.customer; com.company.customer.orders; com.company.customer.info; com.company.sales; com.company.sales.leads; com.company.sales.closed; com.company.orders; com.company.orders.pending; com.company.orders.shipped. To modularize this jar file into three modules, customer, sales, and orders, which module-info.java would be correct?

A)

```

module com.company.customer {
    opens com.company.customer;
}
module com.company.sales{
    opens com.company.sales;
}
module com.company.orders {
    opens com.company.orders;
}

```

B)

```

module com.company.customer {
    exports com.company.customer;
}
module com.company.sales{
    exports com.company.sales;
}
module com.company.orders{
    exports com.company.orders;
}

```

C)

```

module com.company.customer {
    requires com.company.customer;
}
module com.company.sales{
    requires com.company.sales;
}
module com.company.orders {
    requires com.company.orders;
}

```

D)



```
module com.company.customer {
    provides com.company.customer;
}
module com.company.sales{
    provides com.company.sales;
}
module com.company.orders {
    provides com.company.orders;
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** C

#### NEW QUESTION 71

Given the code fragment:

```
int[] secA = { 2, 4, 6, 8, 10 };
int[] secB = { 2, 4, 8, 6, 10 };
int res1 = Arrays.mismatch(secA, secB);
int res2 = Arrays.compare(secA, secB);
System.out.print(res1 + " : " + res2);
```

What is the result?

- A. -1 : 2
- B. 2 : -1
- C. 2 : 3
- D. 3 : 0

**Answer:** B

#### NEW QUESTION 72

Given:

```
String[][] arr = {
    {"Red", "White"},
    {"Black"},
    {"Blue", "Yellow", "Green", "Violet"}
};
for(int row = 0; row < arr.length; row++) {
    int column = 0;
    for(; column < arr[row].length; column++) {
        System.out.println "[" + row + ", " + column + "] = " + arr[row][column]);
    }
}
```

What is the result?

- A. [0,0] = Red[0,1] = White[1,0] = Black[1,1] = Blue[2,0] = Yellow[2,1] = Green[3,0] = Violet
- B. [0,0] = Red[1,0] = Black[2,0] = Blue
- C. java.lang.ArrayIndexOutOfBoundsException thrown
- D. [0,0] = Red[0,1] = White[1,0] = Black[2,0] = Blue[2,1] = Yellow[2,2] = Green[2,3] = Violet

**Answer:** D

**Explanation:**



```
Console 1 Console 2 Console 3
[0,0] =Red
[0,1] =White
[1,0] =Black
[2,0] =Blue
[2,1] =Yellow
[2,2] =Green
[2,3] =Violet
Completed with exit code: 0
```

**NEW QUESTION 77**

Which two describe reasons to modularize the JDK? (Choose two.)

- A. easier to understand the Java language
- B. improves security and maintainability
- C. easier to expose implementation details
- D. improves application robustness
- E. easier to build a custom runtime linking application modules and JDK modules

**Answer:** BD

**NEW QUESTION 79**

Given the code fragment:

```
int x = 0;
do {
    x++;
    if (x == 1) {
        continue;
    }
    System.out.println(x);
} while(x < 1);
```

What is the result?

- A. 01
- B. 1
- C. The program prints nothing.
- D. It prints 1 in the infinite loop.

**Answer:** D

**NEW QUESTION 84**

Given:

```
public class Hello {
    class Greeting {
        void sayHi() {
            System.out.println("Hello world");
        }
    }
    public static void main(String... args) {
        // Line 1
    }
}
```

What code must you insert on Line 1 to enable the code to print Hello world?

- A. Hello.Greeting myG = new Hello.Greeting() myG.sayHi();
- B. Hello myH = new Hello();Hello.Greeting myG = myH.new Greeting(); myG.sayHi();
- C. Hello myH = new Hello();Hello.Greeting myG = myH.new Hello.Greeting(); myG.sayHi();
- D. Hello myH = new Hello(); Greeting myG = new Greeting(); myG.sayHi ();

**Answer:** B

**NEW QUESTION 85**

Which code fragment prints 100 random numbers?

- A. 

```
var r= new Random();
new DoubleStream(r::nextDouble).limit(100).forEach(System.out::print);
```
- B. 

```
DoubleStream.generate(Random::nextDouble)
    .limit (100).forFach(System.out::print);
```
- C. 

```
Doublestream.generate(Random.nextDouble).limit(100).forEach(System.out.print);
```
- D. 

```
var r = new Random(); DoubleStream.generate(r::nextDouble).limit(100).forEach(System.out::print);
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** D

**NEW QUESTION 87**

Given:

```
var i = 10;
var j = 5;
i += (j * 5 + j) / i - 2;
System.out.println(i);
```

What is the result?

- A. 5
- B. 3
- C. 23
- D. 25
- E. 11

**Answer: E****NEW QUESTION 91**

Which two statements are correct about modules in Java? (Choose two.)

- A. java.base exports all of the Java platforms core packages.
- B. module-info.java can be placed in any folder inside module-path.
- C. A module must be declared in module-info.java file.
- D. module-info.java cannot be empty.
- E. By default, modules can access each other as long as they run in the same folder.

**Answer: AC****NEW QUESTION 94**

Given:

```
String originalPath = "data\\projects\\a-project\\..\\..\\another-project"; Path path = Paths.get(originalPath); System.out.print(path.normalize());
```

What is the result?

- A. data\\another-project
- B. data\\projects\\a-project\\another-project
- C. data\\projects\\a-project\\..\\..\\another-project
- D. data\\projects\\a-project\\..\\..\\another-project

**Answer: D****Explanation:**

```
1 import java.util.*;
2 import java.io.*;
3 import java.nio.file.*;
4
5 public class Test {
6
7     public static void main(String[] args) {
8         String originalPath = "data\\projects\\a-project\\..\\..\\another-project";
9         Path path = Paths.get(originalPath);
10        System.out.print(path.normalize());
11    }
12 }
```

Execute Mode, Version, Inputs &amp; Arguments

JDK 11.0.4



Interactive

Stdin Input

CommandLine Arguments



Execute

**Result****CPU Time: 0.19 sec(s), Memory: 31984 kilobyte(s)****data\\projects\\a-project\\..\\..\\another-project**

**NEW QUESTION 95**

Which two are successful examples of autoboxing? (Choose two.)

- A. String a = "A";
- B. Integer e = 5;
- C. Float g = Float.valueOf(null);
- D. Double d = 4;
- E. Long c = 23L;
- F. Float f = 6.0;

**Answer:** AB

**NEW QUESTION 98**

Given:

```
public class Main {  
    public static void main(String[] args) {  
        Consumer consumer = msg -> System.out::print; // line 1  
        consumer.accept("Hello Lambda !");  
    }  
}
```

This code results in a compilation error.

Which code should be inserted on line 1 for a successful compilation?

- A. Consumer consumer = msg -> { return System.out.print(msg); };
- B. Consumer consumer = var arg > {System.out.print(arg);};
- C. Consumer consumer = (String args) > System.out.print(args);
- D. Consumer consumer = System.out::print;

**Answer:** D

**Explanation:**

```
1  import java.util.*;  
2  import java.io.*;  
3  import java.nio.file.*;  
4  import java.util.List;  
5  import java.util.function.Consumer;  
6  
7  public class Main {  
8  
9  public static void main(String[] args) {  
10     Consumer consumer = System.out::print;  
11     consumer.accept("Hello Lambda !");  
12 }  
13 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.16 sec(s), Memory: 32896 kilobyte(s)

Hello Lambda !

**NEW QUESTION 102**

Given:



```
public interface A {
    public Iterable a();
}
public interface B extends A {
    public Collection a();
}
public interface C extends A {
    public Path a();
}
public interface D extends B, C {
}
```

Why does D cause a compilation error?

- A. D inherits a() only from C.
- B. D inherits a() from B and C but the return types are incompatible.
- C. D extends more than one interface.
- D. D does not define any method.

**Answer:** B

#### NEW QUESTION 105

Which two statements are true about Java modules? (Choose two.)

- A. Modular jars loaded from --module-path are automatic modules.
- B. Any named module can directly access all classes in an automatic module.
- C. Classes found in -classpath are part of an unnamed module.
- D. Modular jars loaded from -classpath are automatic modules.
- E. If a package is defined in both the named module and the unnamed module, then the package in the unnamed module is ignored.

**Answer:** AC

#### NEW QUESTION 108

Given:

```
package test;
import java.time.*;
public class Diary {
    private LocalDate now = LocalDate.now();
    public LocalDate getDate() {
        return now;
    }
}
```

and

```
package test;
public class Tester {
    public static void main(String[] args) {
        Diary d = new Diary();
        System.out.println(d.getDate());
    }
}
```

Which statement is true?

- A. Class Tester does not need to import java.time.LocalDate because it is already visible to members of the package test.
- B. All classes from the package java.time
- C. are loaded for the class Diary.
- D. Only LocalDate class from java.time package is loaded.
- E. Tester must import java.time.LocalDate in order to compile.

**Answer:** A

#### NEW QUESTION 111

Given:

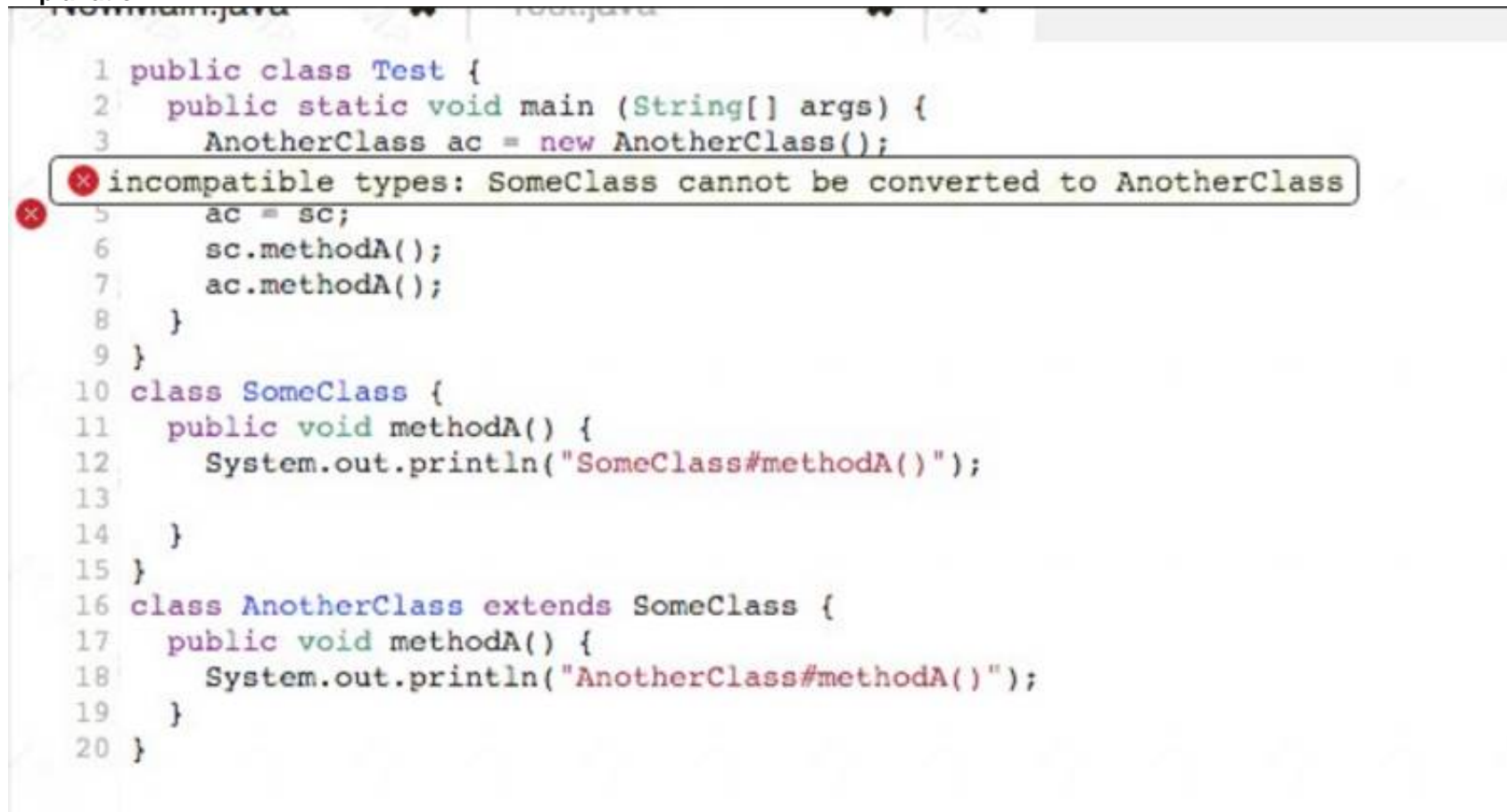
```
public class Test {
    public static void main(String[] args) {
        AnotherClass ac = new AnotherClass();
        SomeClass sc = new AnotherClass();
        ac = sc;
        sc.methodA();
        ac.methodA();
    }
}
class SomeClass {
    public void methodA() {
        System.out.println("SomeClass#methodA()");
    }
}
class AnotherClass extends SomeClass {
    public void methodA() {
        System.out.println("AnotherClass#methodA()");
    }
}
```

What is the result?

- A. A ClassCastException is thrown at runtime.
- B. AnotherClass#methodA()AnotherClass#methodA()
- C. The compilation fails.
- D. SomeClass#methodA()AnotherClass#methodA()
- E. AnotherClass#methodA()SomeClass#methodA()
- F. SomeClass#methodA()SomeClass#methodA()

**Answer: C**

**Explanation:**



```
1 public class Test {
2     public static void main (String[] args) {
3         AnotherClass ac = new AnotherClass();
4
5         ac = sc;
6         sc.methodA();
7         ac.methodA();
8     }
9 }
10 class SomeClass {
11     public void methodA() {
12         System.out.println("SomeClass#methodA()");
13     }
14 }
15 }
16 class AnotherClass extends SomeClass {
17     public void methodA() {
18         System.out.println("AnotherClass#methodA()");
19     }
20 }
```

#### NEW QUESTION 116

Given:

```
public class Main {
    public static void main(String[] args) {
        try (BufferedReader br = new BufferedReader(new InputStreamReader(System.in));) {
            String input = br.readLine();
            System.out.println ("Input String was: " + input);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Which is true?

- A. System.out is the standard output stream
- B. The stream is open only when System.out is called.
- C. System.in cannot reassign the other stream.
- D. System.out is an instance of java.io.OutputStream by default.
- E. System.in is the standard input stream
- F. The stream is already open.

**Answer:** D

#### NEW QUESTION 118

Given:

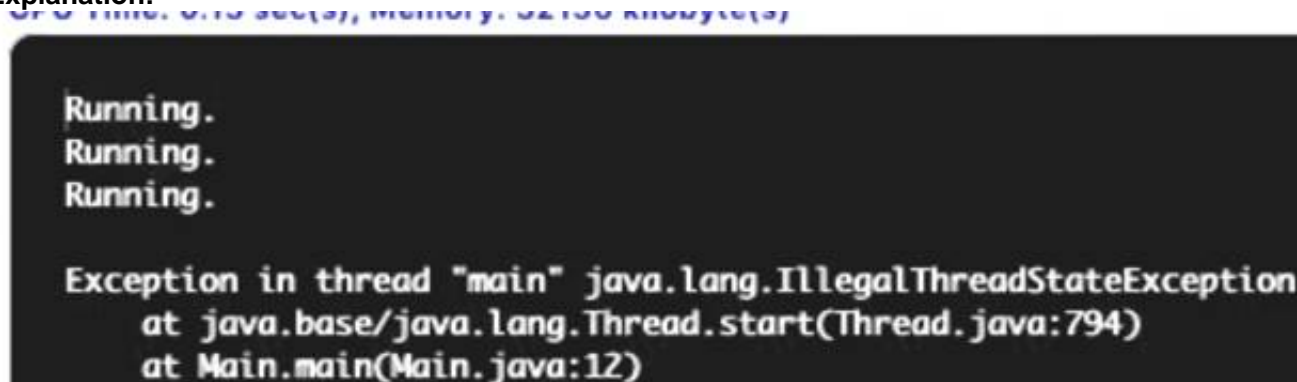
```
public class Main {  
    public static void main(String[] args) {  
        Thread t1 = new Thread(new MyThread());  
        Thread t2 = new Thread(new MyThread());  
        Thread t3 = new Thread(new MyThread());  
  
        t1.start();  
        t2.run();  
        t3.start();  
  
        t1.start();  
    }  
}  
class MyThread implements Runnable {  
    public void run() {  
        System.out.println("Running.");  
    }  
}
```

Which one is correct?

- A. An IllegalStateException is thrown at run time.
- B. Three threads are created.
- C. The compilation fails.
- D. Four threads are created.

**Answer:** A

**Explanation:**



```
Running.  
Running.  
Running.  
  
Exception in thread "main" java.lang.IllegalThreadStateException  
    at java.base/java.lang.Thread.start(Thread.java:794)  
    at Main.main(Main.java:12)
```

#### NEW QUESTION 119

Given:



```
import java.util.List;
import java.util.function.BinaryOperator;
public class Main {
    public static void main(String... args) {
        List<Employee> list = List.of(new Employee("John", 80000.0), new Employee("Scott",
90000.0));
        double starts = 0.0;
        double ratio = 1.0;
        BinaryOperator<Double> bo = (a, b) -> a + b;
double totalSalary = list.stream().map(e -> e.getSalary() * ratio).reduce(starts, bo);
// line 1
        System.out.println("Total salary = " + totalSalary);
    }
}

class Employee {
    String name;
    double salary;
    public Employee(String name, double salary) {
        this.name = name;
        this.salary = salary;
    }
    public String getName() { return name; }
    public double getSalary() { return salary; }
}
```

Which statement is equivalent to line 1?

- A. double totalSalary = list.stream().map(e -> e.getSalary() \* ratio).reduce(bo).ifPresent(p -> p.doubleValue());
- B. double totalSalary = list.stream().mapToDouble(e -> e.getSalary() \* ratio).sum;
- C. double totalSalary = list.stream().map(Employee::getSalary \* ratio).reduce(bo).orElse(0.0);
- D. double totalSalary = list.stream().mapToDouble(e -> e.getSalary() \* ratio).reduce(starts, bo);

**Answer: C**

**Explanation:**

The screenshot shows an IDE with two tabs: Employee.java and Main.java. The Main.java tab is active, displaying the same code as in the question. Below the code editor, the console output is visible, showing "Total salary = 170000.0" and "Completed with exit code: 0".

## NEW QUESTION 121

Given:



```
public class Person {
    private String name;
    public Person(String name) {
        this.name = name;
    }
    public String toString() {
        return name;
    }
}
```

and

```
public class Tester {
    public static void main(String[] args) {
        Person p = null;
        checkPerson(p);
        System.out.println(p);
        p = new Person("Mary");
        checkPerson(p);
        System.out.println(p);
    }
    public static Person checkPerson(Person p) {
        if (p == null) {
            p = new Person("Joe");
        }else{
            p = null;
        }
        return p;
    }
}
```

What is the result?

- A. JoeMarry
- B. Joenull
- C. nullnull
- D. nullMary

**Answer: D**

**Explanation:**



```
Console 1
null
Mary

Console 2

Console 3

Completed with exit code: 0
```

## NEW QUESTION 122

Given:

```
public class Main {
    public static void main(String[] args) {
        Optional<String> value = createValue();
        String str = value.orElse ("Duke");
        System.out.println(str);
    }
    static Optional<String> createValue() {
        String s = null;
        return Optional.ofNullable(s);
    }
}
```

What is the output?

- A. null
- B. A NoSuchElementException is thrown at run time.
- C. Duke
- D. A NullPointerException is thrown at run time.

**Answer: C**

**Explanation:**

```

14
15 ▾ public class Main {
16 ▾     public static void main(String[] args) {
17         Optional<String> value = createValue();
18         String str = value.orElse ("Duke");
19         System.out.println(str);
20     }
21 ▾     static Optional<String> createValue() {
22         String s = null;
23         return Optional.ofNullable(s);
24     }
25 }
26
result
CPU Time: 0.15 sec(s), Memory: 32572 kilobyte(s)
Duke

```

#### NEW QUESTION 127

Given:

```

public class Foo {
    private void print() {
        System.out.println("Bonjour le monde!");
    }
    public void foo() {
        print();
    }
}

public class Bar extends Foo {
    private void print() {
        System.out.println("Hello world!");
    }
    public void bar() {
        print();
    }
    public static void main(String... args) {
        Bar b = new Bar();
        b.foo();
        b.bar();
    }
}

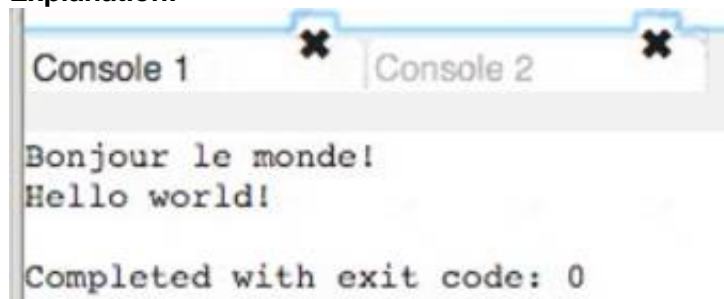
```

What is the output?

- A. Hello world!Bonjour le monde!
- B. Hello world!Hello world!
- C. Bonjour le monde!Hello world!
- D. Bonjour le monde!Bonjour le monde!

**Answer:** C

**Explanation:**



#### NEW QUESTION 131

Given: Automobile.java

```
public abstract class Automobile { //line 1
    abstract void wheels();
}

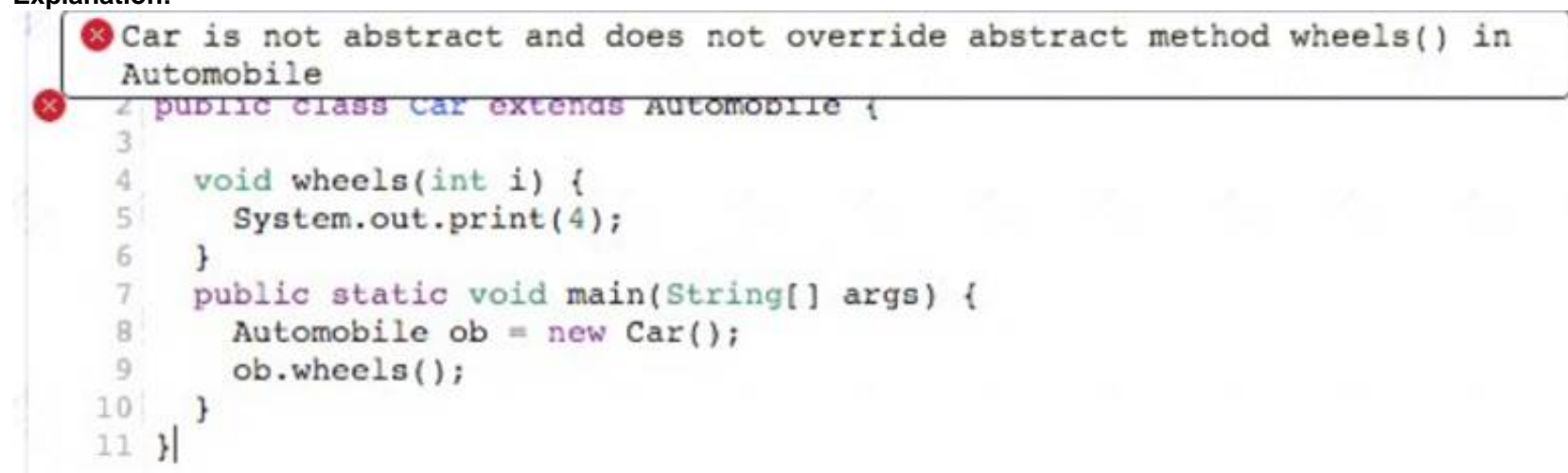
Car.java
public class Car extends Automobile {
    // line 2
    void wheels(int i) { // line 3
        System.out.print(4);
    }
    public static void main(String[] args) {
        Automobile ob = new Car(); // line 4
        ob.wheels();
    }
}
```

What must you do so that the code prints 4?

- A. Remove the parameter from wheels method in line 3.
- B. Add @Override annotation in line 2.
- C. Replace the code in line 2 with Car ob = new Car();
- D. Remove abstract keyword in line 1.

**Answer: B**

**Explanation:**



### NEW QUESTION 133

Given:

```
public interface EulerInterface {
    double getEulerValue();
}

public class EulerLambda {
    public static void main(String[] args) {
        EulerInterface myEulerInterface;
        myEulerInterface = () -> "2.71828";
        System.out.println("Value of Euler = " + myEulerInterface.getEulerValue());
    }
}
```

What is the result?

- A. It throws a runtime exception.
- B. Value of Euler = 2.71828
- C. The code does not compile.
- D. Value of Euler = "2.71828"

**Answer: C**

### NEW QUESTION 136

Which two statements independently compile? (Choose two.)

- A. List<? super Short> list = new ArrayList<Number>();
- B. List<? super Number> list = new ArrayList<Integer>();
- C. List<? extends Number> list = new ArrayList<Byte>();
- D. List<? extends Number> list = new ArrayList<Object>();
- E. List<? super Float> list = new ArrayList<Double>();

**Answer: AC**

**Explanation:**

```

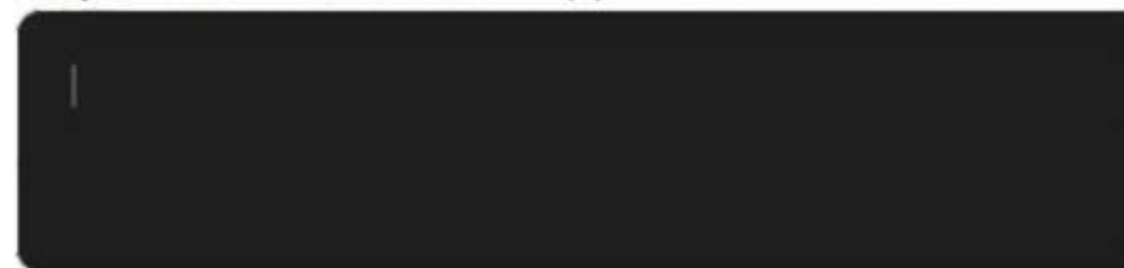
1  import java.util.*;
2  import java.text.*;
3  import java.io.*;
4  import java.lang.Thread;
5  import java.util.ArrayList;
6  import java.util.LinkedList;
7  import java.util.List;
8  import java.util.function.Consumer;
9  import java.util.stream.Stream;
10 import java.util.stream.IntStream;
11 import java.util.Optional;
12
13 public class Intel {
14     public static void main (String[] args) {
15         List<? extends Number> list = new ArrayList<Byte>()
16     }
17 }

```



**Result**

compiled and executed in 1.173 sec(s)



**NEW QUESTION 141**

Given:

/code/a/Test.java containing:

```

package a;
import b.Best;
public class Test {
    public static void main(String[] args) {
        Best b = new Best();
    }
}

```

and

/code/b/Best.java containing: package b;

public class Best { }

Which is the valid way to generate bytecode for all classes?

- A. java /code/a/Test.java
- B. javac -d /code /code/a/Test
- C. java /code/a/Test.java /code/b/Best.java
- D. java -cp /code a.Test
- E. javac -d /code /code/a/Test.java /code/b/Best.java
- F. javac -d /code /code/a/Test.java

**Answer: E**

**NEW QUESTION 145**

Given:

```

public class X {
}

```

and

```

public final class Y extends X {
}

```



What is the result of compiling these two classes?

- A. The compilation fails because there is no zero args constructor defined in class X.
- B. The compilation fails because either class X or class Y needs to implement the toString() method.
- C. The compilation fails because a final class cannot extend another class.
- D. The compilation succeeds.

**Answer: B**

**Explanation:**

```
13
14 public class Main {
15     public static void main (String[] args) {
16         public class X {
17
18         }
19
20     public final class Y extends X {
21
22     }
23 }
24
```

#### NEW QUESTION 149

Given:

```
import java.util.*;
public class Foo {
    public List<Number> foo(Set<CharSequence> m) { ... }
}
```

and

```
import java.util.*;
public class Bar extends Foo {
    //line 1
}
```

Which two statements can be added at line 1 in Bar to successfully compile it? (Choose two.)

- A. public List<Integer> foo(Set<CharSequence> m) { ... }
- B. public ArrayList<Number> foo(Set<CharSequence> m) { ... }
- C. public List<Integer> foo(TreeSet<String> m) { ... }
- D. public List<Integer> foo(Set<String> m) { ... }
- E. public List<Object> foo(Set<CharSequence> m) { ... }
- F. public ArrayList<Integer> foo(Set<String> m) { ... }

**Answer: BC**

#### NEW QUESTION 151

Given:

```
public class Employee {
    private String name;
    private LocalDate birthday;
    // the constructors, getters, and setters methods go here
}
```

and

```
List<Employee> roster = new ArrayList<>();
// ...
Predicate<Employee> y = (Employee e) -> e.getBirthday()
    .isBefore(IsoChronology.INSTANCE.date(1989, 1, 1));
Set<String> s1 = roster.stream()
// Line 1
```

Which code fragment on line 1 makes the s1 set contain the names of all employees born before January 1, 1989?

- A. `.collect(Collectors.partitioningBy(y))  
.get(true)  
.stream()  
.map(Employee::getName)  
.collect(Collectors.toCollection(TreeSet::new));`
- B. `.collect(Collectors.partitioningBy(y))  
.get(true)  
.map(Employee::getName)  
.collect(Collectors.toSet());`
- C. `.collect(Collectors.partitioningBy(y, Collectors.mapping(  
Employee::getName, Collectors.toSet())));`
- D. `.collect(Collectors.partitioningBy(y, Collectors.groupingBy(  
Employee::getName, Collectors.toCollection(TreeSet::new))));`

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** B

#### NEW QUESTION 156

Given:

```
public class Test {
    private int sum;
    public int compute() {
        int x = 0;
        while(x < 3) {
            sum += x++;
        }
        return sum;
    }
    public static void main(String[] args) {
        Test t = new Test();
        int sum = t.compute();
        sum = t.compute();
        t.compute();
        System.out.println(sum);
    }
}
```

What is the result?

- A. 9
- B. An exception is thrown at runtime.
- C. 3
- D. 6

**Answer:** D

**Explanation:**



#### NEW QUESTION 157

Given:

```
public class Main {  
    public static void main(String[] args) {  
        var numbers = List.of(1,2,3,4,5,6,7,8,9,10);  
        Optional<Integer> result = numbers.stream().filter(x -> x % 3 != 0).reduce((i, j)  
-> i + j);  
        result.ifPresent(System.out::print); // line 1  
    }  
}
```

Which is true about line 1?

- A. If the value is not present, a NoSuchElementException is thrown at run time.
- B. It always executes the System.out::print statement.
- C. If the value is not present, a NullPointerException is thrown at run time.
- D. If the value is not present, nothing is done.

**Answer: D**

**Explanation:**

```
1  import java.util.*;  
2  import java.io.*;  
3  import java.lang.Thread;  
4  import java.util.ArrayList;  
5  import java.util.LinkedList;  
6  import java.util.List;  
7  import java.util.function.Consumer;  
8  import java.util.stream.Stream;  
9  import java.util.stream.IntStream;  
10 import java.util.Optional;  
11  
12  
13 public class Main {  
14     public static void main(String[] args) {  
15         var numbers = List.of(1,2,3,4,5,6,7,8,9,10);  
16         Optional<Integer> result = numbers.stream().filter (x -> x % 3 != 0).reduce( (i, j) -> i + j);  
17     }  
18 }  
19 }
```

Result

CPU Time: 0.18 sec(s), Memory: 33380 kilobyte(s)

JDoodle in Action.... Running the program...

#### NEW QUESTION 162

Given:

```
class Super {  
    static String greeting() { return "Good Night"; }  
    String name() { return "Harry"; }  
}
```

and

```
class Sub extends Super {  
    static String greeting() { return "Good Morning"; }  
    String name() { return "Potter"; }  
}
```

and

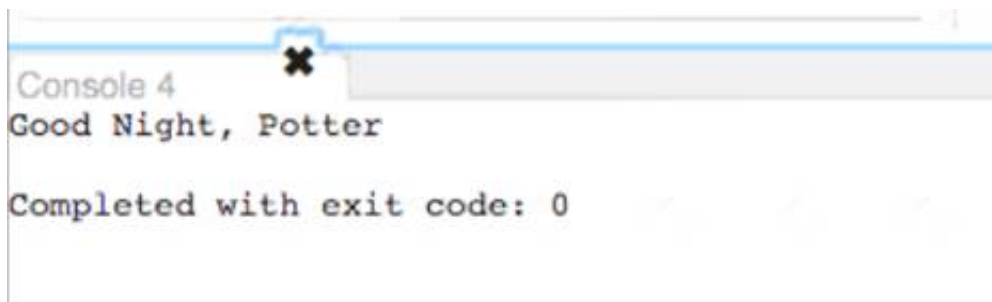
```
class Test {  
    public static void main(String[] args) {  
        Super s = new Sub();  
        System.out.println(s.greeting() + ", " + s.name());  
    }  
}
```

What is the result?

- A. Good Morning, Potter
- B. Good Night, Potter
- C. Good Morning, Harry
- D. Good Night, Harry

**Answer: B**

**Explanation:**



```
Console 4
Good Night, Potter

Completed with exit code: 0
```

**NEW QUESTION 164**

Given:

```
List<String> list = ... ;
list.forEach( x -> { System.out.println(x); } );
```

What is the type of x?

- A. char
- B. List<Character>
- C. String
- D. List<String>

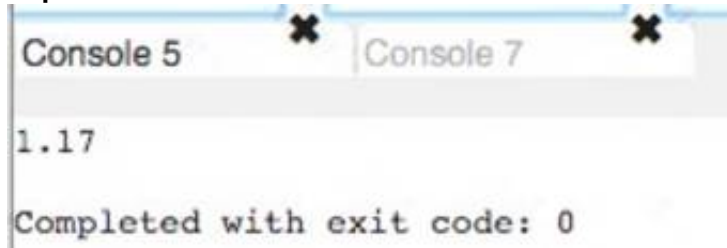
**Answer:** C**NEW QUESTION 166**

Given:

```
public class Tester {
    public static void main(String[] args) {
        byte x = 7, y = 6;
        // line 1
        System.out.println(z);
    }
}
```

Which expression when added at line 1 will produce the output of 1.17?

- A. float z = (float)(Math.round((float)x/y\*100)/100);
- B. float z = Math.round((int)(x/y),2);
- C. float z = Math.round((float)x/y,2);
- D. float z = Math.round((float)x/y\*100)/(float)100;

**Answer:** D**Explanation:**

```
Console 5 | Console 7
1.17

Completed with exit code: 0
```

**NEW QUESTION 170**

Given:

```
public class Sportscar extends Automobile{
    private float turbo;
    ....
    public void setTurbo (float turbo){
        this.turbo = turbo;
    }
}
```

What is known about the Sportscar class?

- A. The Sportscar class is a subclass of Automobile and inherits its methods.
- B. The Sportscar subclass cannot override setTurbo method from the superclass Automobile.
- C. The Sportscar class is a superclass that has more functionality than the Automobile class.
- D. The Sportscar class inherits the setTurbo method from the superclass Automobile.

**Answer:** A**NEW QUESTION 173**



Given:

```
public class Hello {  
    public static void main(String[] args) {  
        System.out.println(args[0]+args[1]+args[2]);  
    }  
}
```

executed using command:

java Hello "Hello World" Hello World What is the output?

- A. An exception is thrown at runtime.
- B. Hello WorldHello World
- C. Hello World Hello World
- D. Hello WorldHelloWorld
- E. HelloHello WorldHelloWorld

**Answer:** C

#### NEW QUESTION 176

Given:

```
List<Reader> dataFiles = new ArrayList<>();  
File indexFile = new File("MyIndex.idx");  
try (BufferedReader indexReader =  
    new BufferedReader(new FileReader(indexFile))) {  
    for(String file = indexReader.readLine(); file != null;  
        file = indexReader.readLine()) {  
        BufferedReader dataReader = new BufferedReader (  
            new FileReader(new File(file))); // Line 1  
        dataFiles.add(dataReader); // Line 2  
        processData(dataReader); // Line 3  
    }  
} catch (IOException ex) {  
    ...  
}  
finally {  
    for(Reader r : dataFiles) {  
        try {  
            r.close();  
        } catch (IOException ex) {  
            ...  
        } // Line 4  
    }  
}
```

What will secure this code from a potential Denial of Service condition?

- A. After Line 4, add indexReader.close().
- B. On Line 3, enclose processData(dataReader) with try with resources.
- C. After Line 3, add dataReader.close().
- D. On Line 1, use try with resources when opening each dataReader.
- E. Before Line 1, check the size of dataFiles to make sure it does not exceed a threshold.

**Answer:** B

#### NEW QUESTION 179

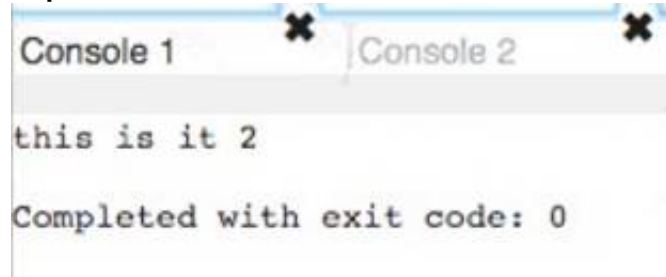
Given:

```
public class Tester {  
    public static void main(String[] args) {  
        String s = "this is it";  
        int x = s.indexOf("is");  
        s.substring(x+3);  
        x = s.indexOf("is");  
        System.out.println(s+" "+x);  
    }  
}
```

What is the result?

- A. is it 1
- B. An IndexOutOfBoundsException is thrown at runtime.
- C. is it 0
- D. this is it 2
- E. this is it 3

**Answer:** D

**Explanation:****NEW QUESTION 183**

Given the contents:

MessageBundle.properties file: message=Hello MessageBundle\_en.properties file: message=Hello (en) MessageBundle\_US.properties file: message=Hello (US)

MessageBundle\_en\_US.properties file: message=Hello (en\_US) MessageBundle\_fr\_FR.properties file: message=Bonjour

and the code fragment: Locale.setDefault(Locale.FRANCE);

Locale currentLocale = new Locale.Builder().setLanguage("en").build();

ResourceBundle messages = ResourceBundle.getBundle("MessageBundle", currentLocale); System.out. println(messages.getString("message"));

Which file will display the content on executing the code fragment?

- A. MessageBundle\_en\_US.properties
- B. MessageBundle\_en.properties
- C. MessageBundle\_fr\_FR.properties
- D. MessageBundle\_US.properties
- E. MessageBundle.properties

**Answer:** C

**NEW QUESTION 187**

Which three annotation uses are valid? (Choose three.)

- A. Function<String, String> func = (@NonNull x) > x.toUpperCase();
- B. var v = "Hello" + (@Interned) "World"
- C. Function<String, String> func = (var @NonNull x) > x.toUpperCase();
- D. Function<String, String> func = (@NonNull var x) > x.toUpperCase();
- E. var myString = (@NonNull String) str;
- F. var obj = new @Interned MyObject();

**Answer:** ACF

**NEW QUESTION 192**

Given:

```
public class Main {
    public static void main(String[] args) {
        for(int i = 0; i < args.length; i++) {
            System.out.println(i + "). " + args[i]);
            switch(args[i]) {
                case "one":
                    continue;
                case "two":
                    i--;
                    continue;
                default:
                    break;
            }
        }
    }
}
```

executed with this command: java Main one two three What is the result?

- A. 0). one
- B. 0). one1). two2). three
- C. The compilation fails.
- D. It creates an infinite loop printing:0). one1). two1). two...
- E. A java.lang.NullPointerException is thrown.

**Answer:** D

**NEW QUESTION 193**

Given:

```
// line 1
List<String> fruits = new ArrayList<>(List.of("apple", "orange", "banana"));
fruits.replaceAll(function);
```

Which statement on line 1 enables this code fragment to compile?

- A. Function function = String::toUpperCase;
- B. UnaryOperator function = s > s.toUpperCase();
- C. UnaryOperator<String> function = String::toUpperCase;
- D. Function<String> function = m > m.toUpperCase();

**Answer:** C

**Explanation:**

```
1
2 import java.io.*;
3 import java.util.*;
4 import java.util.stream.Stream;
5 import java.util.function.Function;
6 import java.util.function.UnaryOperator;
7
8 class Hello {
9     public static void main(String[] args) {
10
11         UnaryOperator<String> function = String::toUpperCase;
12         List<String> fruits = new ArrayList<>(List.of("apple", "orange", "banana"));
13         fruits.replaceAll(function);
14
15     }
16 }
17
```

#### NEW QUESTION 195

Given:

```
public class Price {
    private final double value;
    public Price(String value) {
        this(Double.parseDouble(value));
    }
    public Price(double value) {
        this.value = value;
    }
    public Price () {}
    public double getValue() { return value; }
    public static void main(String[] args) {
        Price p1 = new Price("1.99");
        Price p2 = new Price(2.99);
        Price p3 = new Price();
        System.out.println(p1.getValue()+" "+p2.getValue()+" "+p3.getValue());
    }
}
```

What is the result?

- A. The compilation fail
- B. 1.99,2.99,0
- C. 1.99,2.99,0.0
- D. 1.99,2.99

**Answer:** A

**Explanation:**

```

1
2 public class Price {
3     private final double value;
4     public Price(String value) {
5         this(Double.parseDouble (value));
6     }
7     public Price(double value) {
8         this.value = value;
9     }
10    public Price (){}
11    public double getValue() { return value; }
12    public static void main (String[] args) {
13        Price p1 = new Price("1.99");
14        Price p2 = new Price("2.99");
15        Price p3 = new Price();
16        System.out.println(p1.getValue()+" "+p2.getValue()+" "+p3.getValue());
17    }
18 }

```

✖ variable value might not have been initialized

#### NEW QUESTION 197

Given:

```

import java.util.*;

public class Main {
    static Map<String, String> map = new HashMap<>();
    static List<String> keys =
        new ArrayList<>(List.of("A", "B", "C", "D"));
    static String[] values =
        {"one", "two", "three", "four" };

    static {
        for(var i = 0; i < keys.size(); i++) {
            map.put(keys.get(i), values[i]);
        }
    }

    public static void main(String[] args) {
        keys.clear();
        values = new String[0];
        System.out.println("Map: " + map.size() +
            " Keys: " + keys.size() +
            " Values: " + values.length);
    }
}

```

What is the result?

- A. Map: 0 Keys: 0 Values: 0
- B. The compilation fails.
- C. Map: 4 Keys: 4 Values: 4
- D. Map: 4 Keys: 0 Values: 0
- E. Map: 0 Keys: 4 Values: 4

**Answer: D**

**Explanation:**

Console 1 ✖

```

Map: 4 Keys: 0Values: 0

Completed with exit code: 0

```

#### NEW QUESTION 199

Given:

var fruits = List.of("apple", "orange", "banana", "lemon");

You want to examine the first element that contains the character n. Which statement will accomplish this?

- A. String result = fruits.stream().filter(f -> f.contains("n")).findAny();
- B. fruits.stream().filter(f -> f.contains("n")).forEachOrdered(System.out::print);
- C. Optional<String> result = fruits.stream().filter(f -> f.contains ("n")).findFirst ();
- D. Optional<String> result = fruits.stream().anyMatch(f -> f.contains("n"));



Answer: B

Explanation:

```
1 import java.io.*;
2 import java.util.*;
3 public class abc {
4     public static void main(String[] args) {
5
6         var fruits = List.of("apple", "orange", "banana", "lemon");
7
8         fruits.stream().filter(f -> f.contains("n")).forEachOrdered(System.out::print);
9
10    }
11 }
12
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

Interactive

Stdin Input

CommandLine Arguments

Execute

Result

CPU Time: 0.19 sec(s), Memory: 33200 kilobyte(s)

orangebanana lemon

#### NEW QUESTION 204

Which code is correct?

- A. Runnable r = "Message" > System.out.println();
- B. Runnable r = () > System.out::print;
- C. Runnable r = () -> {System.out.println("Message");};
- D. Runnable r = > System.out.println("Message");
- E. Runnable r = {System.out.println("Message");};

Answer: C

#### NEW QUESTION 206

Given this enum declaration:

```
1. enum Alphabet {
2.     A, B, C
3.
4. }
```

Examine this code: System.out.println(Alphabet.getFirstLetter());  
What code should be written at line 3 to make this code print A?

- A. final String getFirstLetter() { return A.toString(); }
- B. static String getFirstLetter() { return Alphabet.values()[1].toString(); }
- C. static String getFirstLetter() { return A.toString(); }
- D. String getFirstLetter() { return A.toString(); }

Answer: C

#### NEW QUESTION 209

.....

## Thank You for Trying Our Product

\* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

\* One year free update

You can enjoy free update one year. 24x7 online support.

\* Trusted by Millions

We currently serve more than 30,000,000 customers.

\* Shop Securely

All transactions are protected by VeriSign!

**100% Pass Your 1Z0-819 Exam with Our Prep Materials Via below:**

<https://www.certleader.com/1Z0-819-dumps.html>