

# Exam Questions 1Z0-819

Java SE 11 Developer

<https://www.2passeasy.com/dumps/1Z0-819/>



### NEW QUESTION 1

Given:

```
public interface Builder {
    public A build(String str);
}
```

and

```
public class BuilderImpl implements Builder {
    @Override
    public B build(String str) {
        return new B(str);
    }
}
```

Assuming that this code compiles correctly, which three statements are true? (Choose three.)

- A. B cannot be abstract.
- B. B is a subtype of A.
- C. A cannot be abstract.
- D. A cannot be final.
- E. B cannot be final.
- F. A is a subtype of B.

**Answer:** ABD

### NEW QUESTION 2

Given:

```
public class Tester {
    public static void main(String[] args) {
        StringBuilder sb = new StringBuilder(5);
        sb.append("HOWDY");
        sb.insert(0, ' ');
        sb.replace(3, 5, "LL");
        sb.insert(6, "COW");
        sb.delete(2, 7);
        System.out.println(sb.length());
    }
}
```

What is the result?

- A. 4
- B. 3
- C. An exception is thrown at runtime.
- D. 5

**Answer:** D

**Explanation:**

```
6 public class Tester {
7     public static void main(String[] args) {
8         StringBuilder sb = new StringBuilder(5);
9         sb.append("HOWDY");
10        sb.insert(0, ' ');
11        sb.replace(3, 5, "LL");
12        sb.insert(6, "COW");
13        sb.delete(2, 7);
14        System.out.println(sb.length());
15    }
16 }
```

(command line arguments)

COMPILE & EXECUTE

PASTE SOURCE

Successfully compiled /tmp/java\_82Tlan/Tester.java <-- main method

5

### NEW QUESTION 3

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 4

A bookstore's sales are represented by a list of Sale objects populated with the name of the customer and the books they purchased.

```
public class Sale { private String customer;
private List<Book> items;
// constructor, setters and getters not shown
}
public class Book { private String name; private double price;
// constructor, setters and getters not shown
}
```

Given a list of Sale objects, `tList`, which code fragment creates a list of total sales for each customer in ascending order?

- A. 

```
List<String> totalByUser = tList.stream()
    .collect(flatMapping(t -> t.getItems().stream(),
                        groupingBy(Sale::getCustomer,
                                summingDouble(Book::getPrice))))
    .entrySet().stream()
    .sorted(Comparator.comparing(Entry::getValue))
    .collect(mapping(e -> e.getKey() + ":" + e.getValue(),toList()));
```
- B. 

```
List<String> totalByUser = tList.stream()
    .collect(groupingBy(Sale::getCustomer,
                        flatMapping(t -> t.getItems().stream(),
                                summingDouble(Book::getPrice))))
    .sorted(Comparator.comparing(Entry::getValue))
    .collect(mapping(e -> e.getKey() + ":" + e.getValue(),toList()));
```
- C. 

```
List<String> totalByUser = tList.stream()
    .collect(groupingBy(Sale::getCustomer,
                        flatMapping(t -> t.getItems().stream(),
                                summingDouble(Book::getPrice))))
    .entrySet().stream()
    .sorted(Comparator.comparing(Entry::getValue))
    .collect(mapping(e -> e.getKey() + ":" + e.getValue(),toList()));
```
- D. 

```
List<String> totalByUser = tList.stream()
    .collect(flatMapping(t -> t.getItems().stream(),
                        groupingBy(Sale::getCustomer,
                                summingDouble(Book::getPrice))))
    .sorted(Comparator.comparing(Entry::getValue))
    .collect(mapping(e -> e.getKey() + ":" + e.getValue(),toList()));
```

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

**Answer:** C

#### NEW QUESTION 5

Given:

```
import java.time.LocalDate;
import static java.time.DayOfWeek.*;
public class Main {
    public static void main(String[] args) {
        var today = LocalDate.now().with(TUESDAY).getDayOfWeek();
        switch(today) {
            case SUNDAY:
            case SATURDAY:
                System.out.println("Weekend");
                break;
            case MONDAY:    FRIDAY:
                System.out.println("Working");
            default:
                System.out.println("Unknown");
        }
    }
}
```

What is the result?

- A. WorkingUnknown
- B. Unknown
- C. TuesdayUnknown
- D. The compilation fails.
- E. Tuesday
- F. Working

**Answer:** B

**Explanation:**



#### NEW QUESTION 6

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:**



DE

DOWNLOAD ZIP

default

2,5

4,5

4,5

#### NEW QUESTION 7

Given:

```
public class Tester {
    public static void main(String[] args) {
        char letter = 'b';
        int i = 0;
        switch(letter) {
            case 'a':
                i++;
                break;
            case 'b':
                i++;
            case 'c' | 'd': // line 1
                i++;
            case 'e':
                i++;
                break;
            case 'f':
                i++;
                break;
            default:
                System.out.print(letter);
        }
        System.out.println(i);
    }
}
```

What is the result?

- A. b1
- B. 2
- C. b2
- D. 1
- E. b3
- F. 3
- G. The compilation fails due to an error in line 1.

Answer: F

Explanation:

Result

CPU Time: 0.23 sec(s), Memory: 32708 kilobyte(s)

3

#### NEW QUESTION 8

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 9

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 10

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 &amp; Arguments

JDK 11.0.4

## CommandLine Arguments

## Result

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

6104 3

## NEW QUESTION 10

Given:

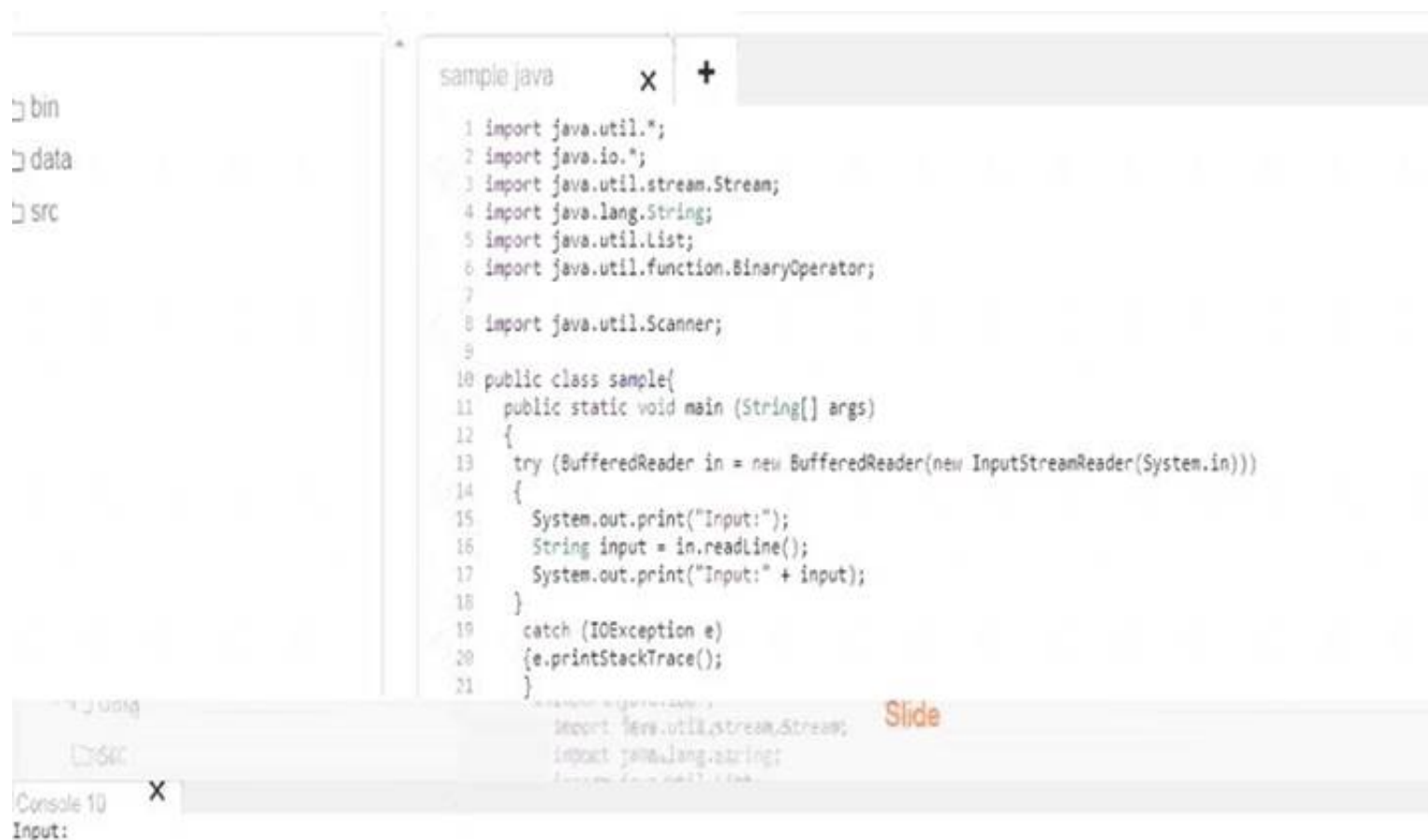
```
public class Main {
    public static void main(String[] args) {
        try(BufferedReader in = new BufferedReader(new InputStreamReader(System.in))) {
            System.out.print("Input: ");
            String input = in.readLine();
            System.out.println("Echo: " + input);
        } catch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

And the command: java Main Helloworld What is the result ?

- A. Input: Echo:
- B. Input: Helloworld Echo: Helloworld
- C. Input:Then block until any input comes from System.in.
- D. Input:Echo: Helloworld
- E. A NullPointerException is thrown at run time.

Answer: C

Explanation:



### NEW QUESTION 13

Given:

```

public class SerializedMessage implements Serializable {
    String message;
    LocalDateTime createdAt;
    transient LocalDateTime updatedAt;
    SerializedMessage(String message) {
        this.message = message;
        this.createdAt = LocalDateTime.now();
    }
    private void readObject (ObjectInputStream in) {
        try {
            in.defaultReadObject();
            this.updatedAt = LocalDateTime.now();
        } catch (IOException | ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
}

```

When is the readObject method called?

- A. before this object is deserialized
- B. after this object is deserialized
- C. before this object is serialized
- D. The method is never called.
- E. after this object is serialized

**Answer: B**

### NEW QUESTION 14

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.");
        }
    }

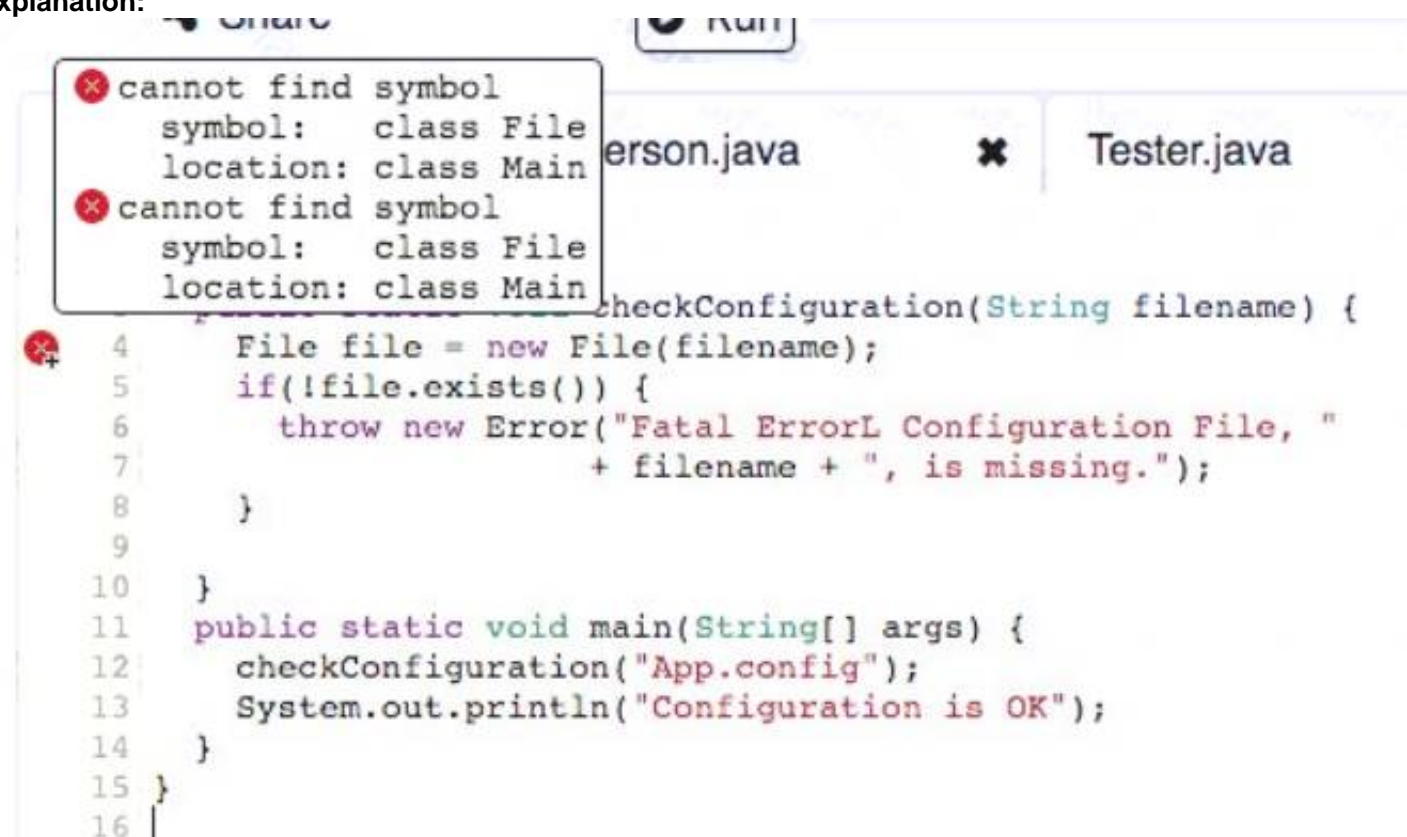
    public static void main(String[] args) {
        checkConfiguration("App.config");
        System.out.println("Configuration is OK");
    }
}
```

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

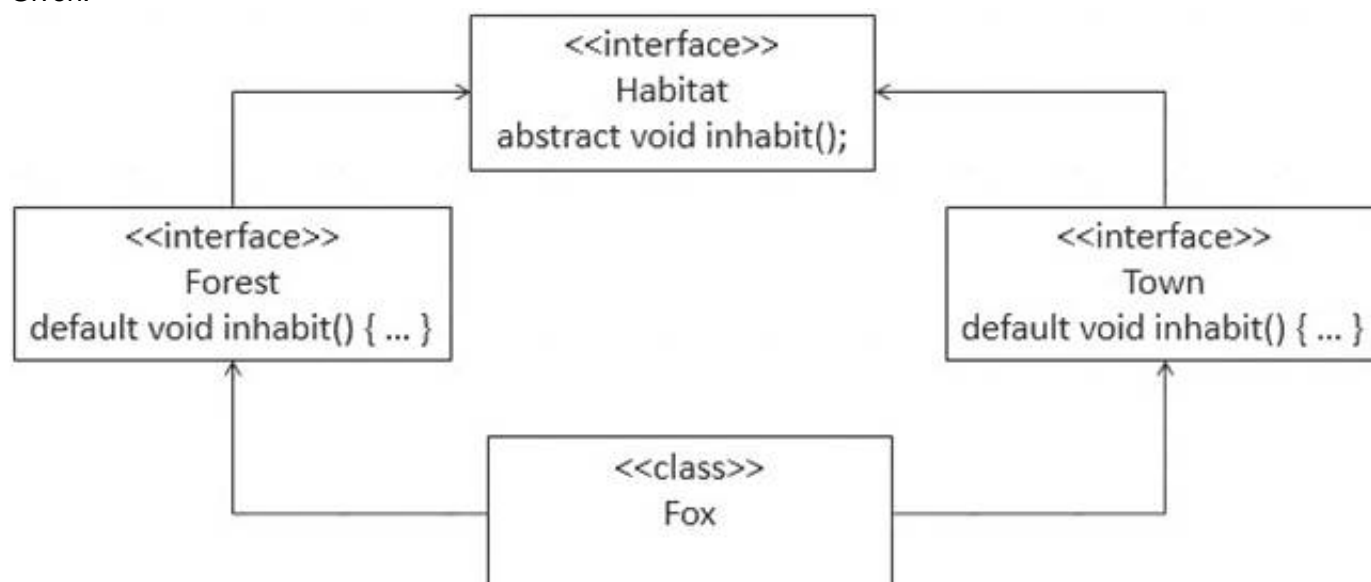
Answer: B

Explanation:



#### NEW QUESTION 17

Given:



Which statement is true about the Fox class?

- A. Fox class does not have to override inhabit method, so long as it does not try to call it.
- B. Fox class does not have to override the inhabit method if Forest and Town provide compatible implementations.
- C. Fox class must implement either Forest or Town interfaces, but not both.
- D. The inhabit method implementation from the first interface that Fox implements will take precedence.
- E. Fox class must provide implementation for the inhabit method.

Answer: B

#### NEW QUESTION 22

Given the code fragment:

```
public static void main(String[] args) {  
    List<Integer> even = List.of();  
    even.add(0, -1);  
    even.add(0, -2);  
    even.add(0, -3);  
    System.out.println(even);  
}
```

What is the output?

- A. The compilation fail
- B. [-1, -2, -3]
- C. [-3, -2, -1]
- D. A runtime exception is thrown.

Answer: D

#### NEW QUESTION 26

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"+ MMMMM yyyy"
- D. "eeee d'th of' MMMMM yyyy"

Answer: B

#### NEW QUESTION 29

Given:

```
public class Test{  
    private int num = 1;  
    private int div = 0;  
  
    public void divide() {  
        try {  
            num = num / div;  
            System.out.print("Exception");  
        }  
        catch(ArithmeticException ae) { num = 100; }  
        catch(Exception e) { num = 200; }  
        finally { num = 300; }  
        System.out.print(num);  
    }  
    public static void main(String args[])  
    {  
        Test test = new Test();  
        test.divide();  
    }  
}
```

What is the output?

- A. 300
- B. Exception
- C. 200
- D. 100

Answer: A

Explanation:

```
1 public class Test{
2     private int num = 1;
3     private int div = 0;
4
5     public void divide() {
6         try {
7             num = num / div;
8             System.out.print("Exception");
9         }
10        catch(ArithmeticException ae) { num = 100; }
11        catch(Exception e) { num = 200; }
12        finally { num = 300; }
13        System.out.print(num);
14    }
15    public static void main(String args[])
16    {
17        Test test = new Test();
18        test.divide();
19    }
20 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

☐ In

CommandLine Arguments

Result

CPU Time: 0.15 sec(s), Memory: 32484 kilobyte(s)

300

#### NEW QUESTION 34

Given:

```
public class X {
    private Collection collection;
    public void set(Collection collection) {
        this.collection = collection;
    }
}
```

and

```
public class Y extends X {
    public void set(Map<String,String> map) {
        super.set(map); // line 1
    }
}
```

Which two lines can replace line 1 so that the Y class compiles? (Choose two.)

- A. map.forEach((k, v) -> set(v));
- B. set(map.values());
- C. super.set(List<String> map)
- D. super.set(map.values());
- E. set(map)

Answer: BD

#### NEW QUESTION 39

Given:

```
public class DNASynth {
    int aCount;
    int tCount;
    int cCount;
    int gCount;

    DNASynth(int a, int tCount, int c, int g){
        // line 1
    }
    int setCCount(int c){
        return c;
    }
    void setGCount(int gCount){
        this.gCount = gCount;
    }
}
```

Which two lines of code when inserted in line 1 correctly modifies instance variables? (Choose two.)

- A. setCCount(c) = cCount;
- B. tCount = tCount;
- C. setGCount(g);
- D. cCount = setCCount(c);
- E. aCount = a;

**Answer:** BE

#### NEW QUESTION 40

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

...

Copy

Result

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

```
apple:APPLE
orange:ORANGE
banana:BANANA
```

#### NEW QUESTION 42

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 46

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 50

Which statement about a functional interface is true?

- A. It must be defined with the public access modifier.
- B. It must be annotated with @FunctionalInterface.
- C. It is declared with a single abstract method.
- D. It is declared with a single default method.
- E. It cannot have any private methods and static methods.

**Answer:** C

#### NEW QUESTION 55

Which two statements correctly describe capabilities of interfaces and abstract classes? (Choose two.)

- A. Interfaces cannot have protected methods but abstract classes can.
- B. Both interfaces and abstract classes can have final methods.
- C. Interfaces cannot have instance fields but abstract classes can.
- D. Interfaces cannot have static methods but abstract classes can.
- E. Interfaces cannot have methods with bodies but abstract classes can.

**Answer:** AC

#### NEW QUESTION 57

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 61

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 & 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 65

Assume ds is a DataSource and the EMP table is defined appropriately.

```

try (Connection conn = ds.getConnection();
     PreparedStatement ps = conn.prepareStatement("INSERT INTO EMP VALUES(?, ?, ?)")) {
    ps.setObject(1, 101, JDBCType.INTEGER);
    ps.setObject(2, "SMITH", JDBCType.VARCHAR);
    ps.setObject(3, "HR", JDBCType.VARCHAR);
    ps.executeUpdate();
    ps.setInt(1, 102);
    ps.setString(2, "JONES");
    ps.executeUpdate();
}
    
```

What does executing this code fragment do?

- A. inserts two rows (101, 'SMITH', 'HR') and (102, 'JONES', NULL)
- B. inserts two rows (101, 'SMITH', 'HR') and (102, 'JONES', 'HR')
- C. inserts one row (101, 'SMITH', 'HR')
- D. throws a SQLException

Answer: C

#### NEW QUESTION 67

Given:

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

and

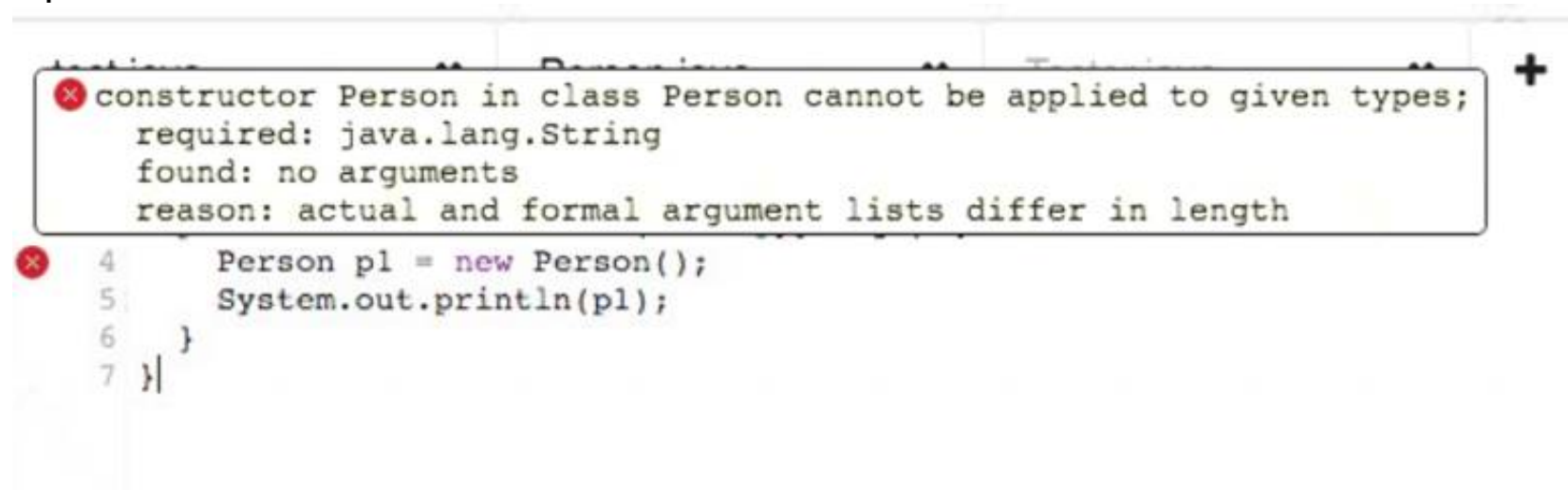
```
public class Tester {
    public static void main(String[] args) {
        Person p1 = new Person(); // line 1
        System.out.println(p1);
    }
}
```

What is the result?

- A. null
- B. Joe Bloggs
- C. The compilation fails due to an error in line 1.
- D. p1

**Answer: C**

**Explanation:**



#### NEW QUESTION 70

Which three guidelines are used to protect confidential information? (Choose three.)

- A. Limit access to objects holding confidential information.
- B. Clearly identify and label confidential information.
- C. Manage confidential and other information uniformly.
- D. Transparently handle information to improve diagnostics.
- E. Treat user input as normal information.
- F. Validate input before storing confidential information.
- G. Encapsulate confidential information.

**Answer: ADF**

#### NEW QUESTION 74

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 77

Which describes an aspect of Java that contributes to high performance?

- A. Java prioritizes garbage collection.
- B. Java has a library of built-in functions that can be used to enable pipeline burst execution.
- C. Java monitors and optimizes code that is frequently executed.
- D. Java automatically parallelizes code execution.



Answer: C

#### NEW QUESTION 82

Given:

```
import java.util.function.BiFunction;
public class Pair<T> {
    final BiFunction<T, T, Boolean> validator;
    T left = null;
    T right = null;
    private Pair() {
        validator=null;
    }
    Pair(BiFunction<T, T, Boolean> v, T x, T y) {
        validator = v;
        set(x, y);
    }
    void set(T x, T y) {
        if (!validator.apply(x, y)) throw new IllegalArgumentException();
        setLeft(x);
        setRight(y);
    }
    void setLeft(T x) {
        left = x;
    }
    void setRight(T y) {
        right = y;
    }
    final boolean isValid() {
        return validator.apply(left, right);
    }
}
```

It is required that if p instanceof Pair then p.isValid() returns true.

Which is the smallest set of visibility changes to insure this requirement is met?

- A. setLeft and setRight must be protected.
- B. left and right must be private.
- C. isValid must be public.
- D. left, right, setLeft, and setRight must be private.

Answer: B

#### NEW QUESTION 85

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 90

Given:

```

1. public class Main {
2.     public static void greet(String... args) {
3.         System.out.print("Hello ");
4.         for (String arg : args) {
5.             System.out.println(arg);
6.         }
7.     }
8.     public static void main(String[] args) {
9.         Main c = null;
10.        c.greet();
11.    }
12. }

```

What is the result?

- A. NullPointerException is thrown at line 4.
- B. NullPointerException is thrown at line 10.
- C. A compilation error occurs.
- D. Hello

Answer: D

Explanation:



Console 4 Console 5

hello

Completed with exit code: 0

#### NEW QUESTION 95

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 97

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 102

Given this requirement:

Module vehicle depends on module part and makes its com.vehicle package available for all other modules. Which module-info.java declaration meets the requirement?

A

```
module vehicle{
    requires part;
    exports com.vehicle;
}
```

B

```
module vehicle {
    requires part;
    uses com.vehicle;
}
```

C

```
module vehicle{
    requires part;
    exports com.vehicle to part;
}
```

D

```
module vehicle {
    requires com.vehicle;
    exports part;
}
```

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

**Answer:** A**NEW QUESTION 104**

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 108**

Given:

```
var data = new ArrayList<>(); data.add("Peter");
data.add(30); data.add("Market Road"); data.set(1, 25); data.remove(2); data.set(3, 1000L); System.out.print(data);
```

What is the output?

- A. [Market Road, 1000]
- B. [Peter, 30, Market Road]
- C. [Peter, 25, null, 1000]
- D. An exception is thrown at run time.

**Answer:** D



#### Explanation:

```

Console 1
Exception in thread "main" java.lang.IndexOutOfBoundsException: Index 3 out of bounds for length 2
    at java.base/jdk.internal.util.Preconditions.outOfBounds(Preconditions.java:64)
    at java.base/jdk.internal.util.Preconditions.outOfBoundsCheckIndex(Preconditions.java:70)
    at java.base/jdk.internal.util.Preconditions.checkIndex(Preconditions.java:248)
    at java.base/java.util.Objects.checkIndex(Objects.java:372)
    at java.base/java.util.ArrayList.set(ArrayList.java:472)
    at abc.main(abc.java:13)

Completed with exit code: 1

```

#### NEW QUESTION 113

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 Console 2 Console 3
null
Mary

Completed with exit code: 0

```

#### NEW QUESTION 118

Which is a proper JDBC URL?

- A. jdbe.mysql.com://localhost:3306/database
- B. http://localhost.mysql.com:3306/database
- C. http://localhostmysql.jdbc:3306/database
- D. jdbc:mysql://localhost:3306/database

**Answer: D**

### NEW QUESTION 120

Given: Automobile.java

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

Car.java

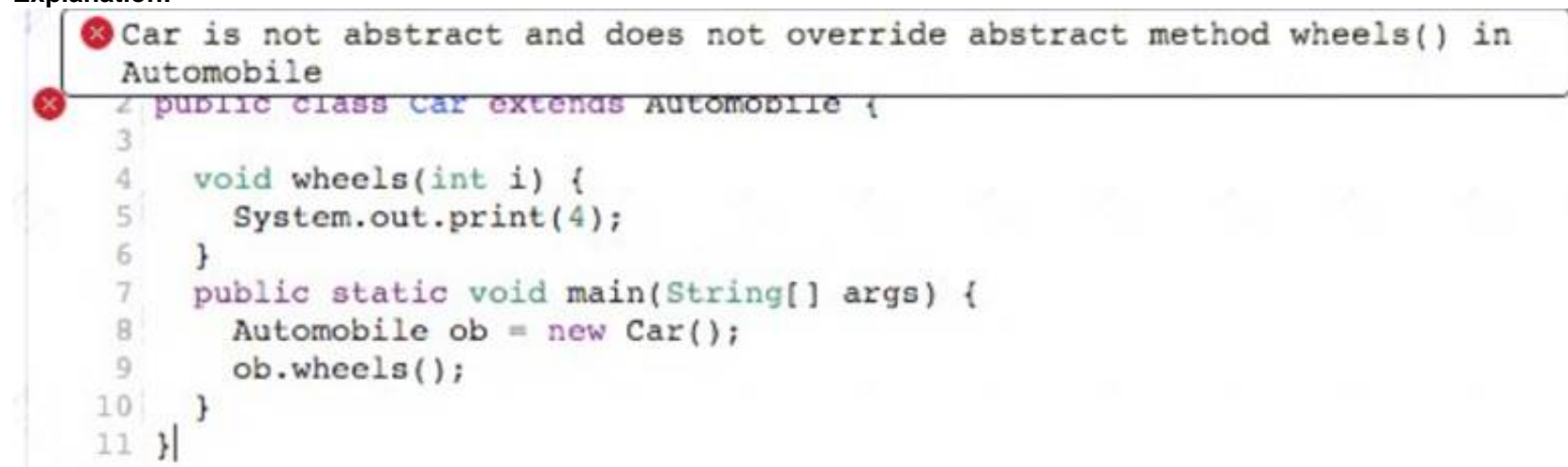
```
public class Car extends Automobile {
    // line 2
    void wheels(int i) { // line 3
        System.out.print(4);
    }
    public static void main(String[] args) {
        Automobile ob = new Car(); // line 4
        ob.wheels();
    }
}
```

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 125

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 }

```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

## Result

compiled and executed in 1.173 sec(s)

## NEW QUESTION 129

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 134

Given the code fragment:

```
Path source = Paths.get("/repo/a/a.txt"); Path destination = Paths.get("/repo"); Files.move(source, destination); // line 1
Files.delete (source); // line 2
```

Assuming the source file and destination folder exist, what is the result?

- A. A java.nio.file.FileAlreadyExistsException is thrown on line 1.
- B. A java.nio.file.NoSuchFileException is thrown on line 2.
- C. A copy of /repo/a/a.txt is moved to the /repo directory and /repo/a/a.txt is deleted.
- D. a.txt is renamed repo.

**Answer:** C

#### NEW QUESTION 137

Given:

```
import java.io.*;
public class Tester {
    public static void main(String[] args) {
        try {
            doA();
            doB();
        } catch(IOException e) {
            System.out.print("c");
            return;
        } finally{
            System.out.print("d");
        }
        System.out.print("f");
    }
    private static void doA() {
        System.out.print("a");
        if (false) {
            throw new IndexOutOfBoundsException();
        }
    }
    private static void doB() throws FileNotFoundException {
        System.out.print("b");
        if (true) {
            throw new FileNotFoundException();
        }
    }
}
```

What is the result?

- A. The compilation fails.
- B. abdf
- C. abd
- D. adf
- E. abcd

**Answer:** E

#### NEW QUESTION 138

Given:

```
enum Color implements Serializable {
    R(1), G(2), B(3);
    int c;
    public Color(int c) {
        this.c = c;
    }
}
```

What action ensures successful compilation?

- A. Replace public Color(int c) with private Color(int c).
- B. Replace int c; with private int c;.
- C. Replace int c; with private final int c;.
- D. Replace enum Color implements Serializable with public enum Color.
- E. Replace enum Color with public enum Color.

**Answer:** A

**Explanation:**



```
1
2 import java.io.*;
3 import java.util.*;
4 class Hello {
5
6
7     enum Color implements Serializable {
8         R(1), G(2), B(3);
9         int c;
10        private Color (int c) {
11            this.c = c;
12        }
13    }
14 }
```

**NEW QUESTION 139**

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 141**

Given:

```
package test.t1;
public class A {
    public int x = 42;
    protected A() {}           // line 1
}
```

and

```
package test.t2;
import test.t1.*;
public class B extends A {
    int x = 17;                 // line 2
    public B() { super(); }     // line 3
}
```

and

```
package test;
import test.t1.*;
import test.t2.*;
public class Tester {
    public static void main(String[] args) {
        A obj = new B();        // line 4
        System.out.println(obj.x); // line 5
    }
}
```

What is the result?

- A. 42
- B. The compilation fails due to an error in line 4.
- C. 17
- D. The compilation fails due to an error in line 3.
- E. The compilation fails due to an error in line 2.
- F. The compilation fails due to an error in line 1.
- G. The compilation fails due to an error in line 5.

**Answer: A**

#### NEW QUESTION 142

var numbers = List.of(0,1,2,3,4,5,6,7,8,9);

You want to calculate the average of numbers. Which two codes will accomplish this? (Choose two.)

- A. double avg = numbers.stream().parallel().averagingDouble(a > a);
- B. double avg = numbers.parallelStream().mapToInt (m > m).average().getAsDouble ();
- C. double avg = numbers.stream().mapToInt (i > i).average().parallel();
- D. double avg = numbers.stream().average().getAsDouble();
- E. double avg = numbers.stream().collect(Collectors.averagingDouble(n > n));

**Answer:** BD

**Explanation:**

```

1
2 import java.io.*;
3 import java.util.*;
4 class Hello {
5     public static void main(String[] args) {
6
7         var numbers = List.of(0,1,2,3,4,5,6,7,8,9);
8         double avg = numbers.parallelStream().mapToInt (m -> m).average().getAsDouble();
9
10    }
11 }
```

#### NEW QUESTION 143

Given:

```

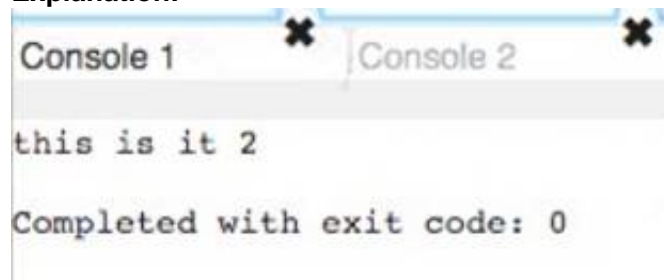
public class Tester {
    public static void main(String[] args) {
        String s = "this is it";
        int x = s.indexOf("is");
        s.substring(x+3);
        x = s.indexOf("is");
        System.out.println(s+" "+x);
    }
}
```

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:**



```

Console 1
Console 2

this is it 2

Completed with exit code: 0
```

#### NEW QUESTION 147

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 150

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);
    }
}

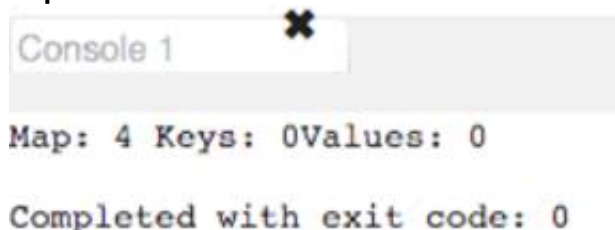
```

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 151

Given:

```
for(var i = 0; i < 10; i++) {  
    switch(i%5) {  
        case 2:  
            i *= i;  
            break;  
        case 3:  
            i++;  
            break;  
        case 1:  
        case 4:  
            i++;  
            continue;  
        default:  
            break;  
    }  
    System.out.print(i + " ");  
    i++;  
}
```

What is the result?

- A. nothing
- B. 10
- C. 0 4 9

**Answer:** A

#### NEW QUESTION 153

Given:

```
public class Foo {  
    public static void main(String... args) {  
        for (var x : args) {  
            System.out.println(x);  
        }  
    }  
}
```

What is the type of the local variable x?

- A. Character
- B. char
- C. String[ ]
- D. String

**Answer:** D

#### NEW QUESTION 154

.....



## THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual 1Z0-819 Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the 1Z0-819 Product From:

<https://www.2passeasy.com/dumps/1Z0-819/>

## Money Back Guarantee

### 1Z0-819 Practice Exam Features:

- \* 1Z0-819 Questions and Answers Updated Frequently
- \* 1Z0-819 Practice Questions Verified by Expert Senior Certified Staff
- \* 1Z0-819 Most Realistic Questions that Guarantee you a Pass on Your FirstTry
- \* 1Z0-819 Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year