

# Oracle

## Exam Questions 1Z0-819

Java SE 11 Developer



### NEW QUESTION 1

Given:

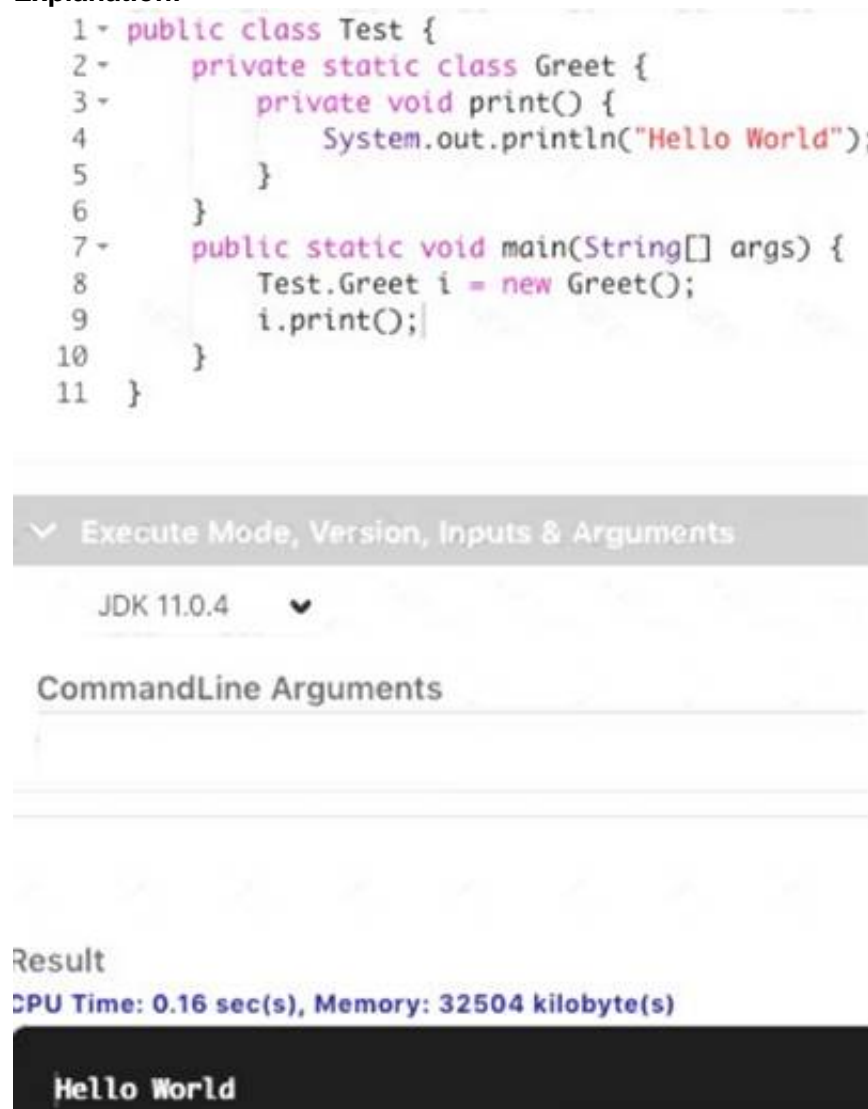
```
1. public class Test {
2.     private static class Greet {
3.         private void print() {
4.             System.out.println("Hello World");
5.         }
6.     }
7.     public static void main(String[] args) {
8.         Test.Greet i = new Greet();
9.         i.print();
10.    }
11. }
```

What is the result?

- A. The compilation fails at line 9.
- B. The compilation fails at line 2.
- C. Hello World
- D. The compilation fails at line 8.

**Answer: C**

**Explanation:**



```
1 public class Test {
2     private static class Greet {
3         private void print() {
4             System.out.println("Hello World");
5         }
6     }
7     public static void main(String[] args) {
8         Test.Greet i = new Greet();
9         i.print();
10    }
11 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.16 sec(s), Memory: 32504 kilobyte(s)

Hello World

### NEW QUESTION 2

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 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:

```
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 6

Which code fragment does a service use to load the service provider with a Print interface?

- A. `private Print print = com.service.Provider.getInstance();  
B. private java.util.ServiceLoader<Print> loader = ServiceLoader.load(Print.class);  
C. private java.util.ServiceLoader<Print> loader = new java.util.ServiceLoader<>();  
D. private Print print = new com.service.Provider.PrintImpl();`

**Answer: B**

#### NEW QUESTION 7

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 8

Given the code fragment:

```
int x = 0;
while(x < 10){
    System.out.print(x++);
}
```

Which "for" loop produces the same output?

A.

```
int b = 0;
for( ; b < 10; ){
    System.out.print(++b);
}
```

B.

```
for(a; a < 10; a++){
    System.out.print(a);
}
```

C.

```
for(int d = 0; d < 10; ){
    System.out.print(d);
    ++d;
}
```

D.

```
for(int c = 0; ; c++){
    System.out.print(c);
    if(c == 10){
        break;
    }
}
```

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

**Answer:** C

#### NEW QUESTION 9

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 10

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 10

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 11

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 15

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 20**

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:

```
public class Main {
    class Student {                                // line 1
        String classname;
        Student(String classname) {                // line 2
            this.classname = classname;
        }
    }
    public static void main(String[] args) {
        var student = new Student("Biology"); // line 3
    }
}
```

Which two independent changes will make the Main class compile? (Choose two.)

- A. Move the entire Student class declaration to a separate Java file, Student.java.
- B. Change line 2 to `public Student(String classname)`.
- C. Change line 1 to `public class Student {`.
- D. Change line 3 to `Student student = new Student("Biology");`.
- E. Change line 1 to `static class Student {`.

**Answer: BD****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     class Student {
15         String classname;
16         public Student (String classname) {
17             this.classname = classname;
18         }
19
20     }
21     public static void main (String[] args) {
22         var student = new Student ("Biology");
23     }
24 }
```

**NEW QUESTION 26**

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 31**

Given the code fragment:

var pool = Executors.newFixedThreadPool(5);

Future outcome = pool.submit(() &gt; 1);

Which type of lambda expression is passed into submit()?

- A. java.lang.Runnable
- B. java.util.function.Predicate
- C. java.util.function.Function
- D. java.util.concurrent.Callable

**Answer: D****NEW QUESTION 36**

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.");
        }
    }

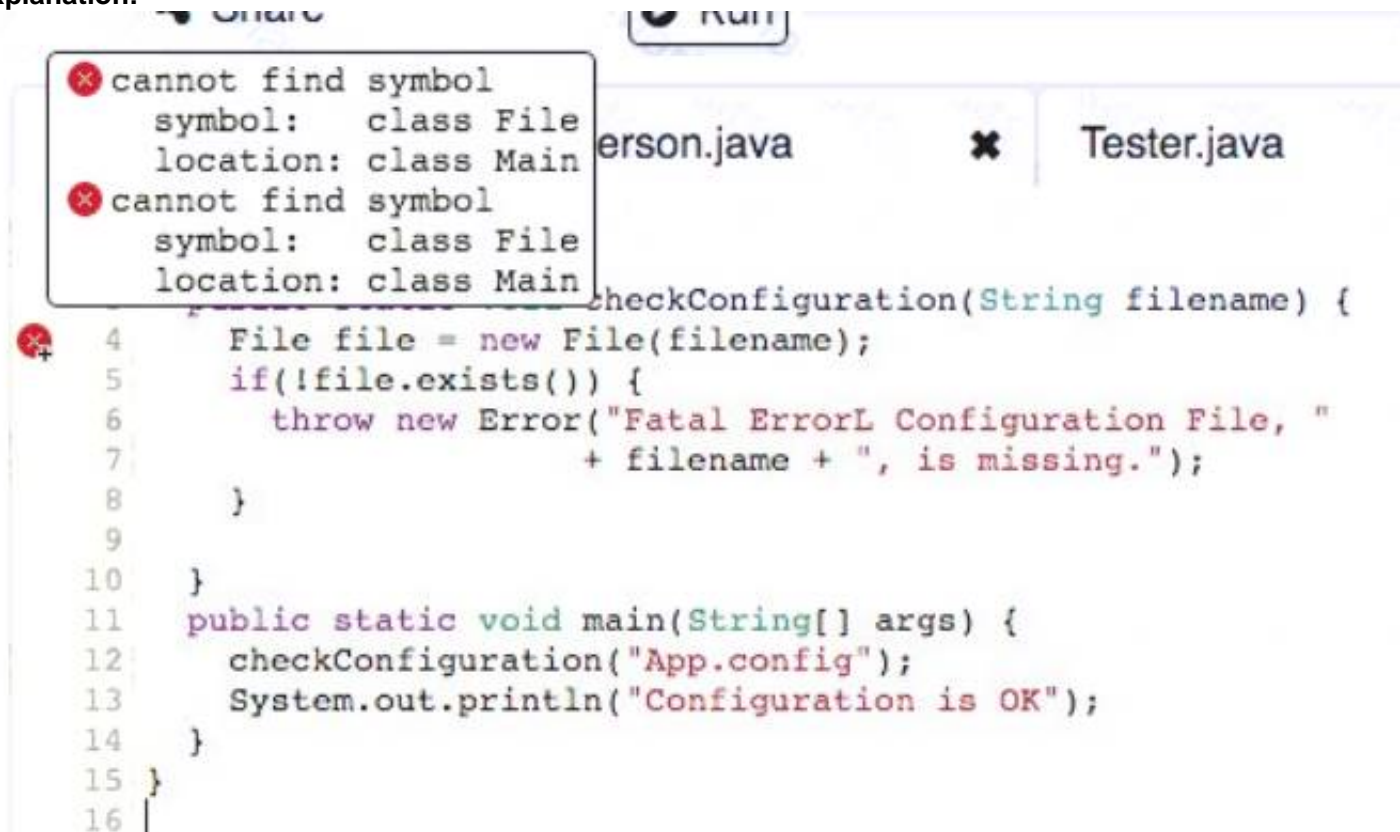
    public static void main(String[] args) {
        checkConfiguration("App.config");
        System.out.println("Configuration is OK");
    }
}
```

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 38

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 41

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 43

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 45

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 49

Given:

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

and

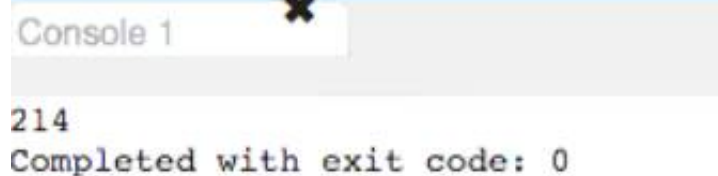
```
public class ConSub extends ConSuper{
    ConSub() {
        this(4);
        System.out.print("3");
    }
    ConSub(int a) {
        System.out.print(a);
    }
    public static void main (String[] args) {
        new ConSub(4);
    }
}
```

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 51

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.enquoteIdentifier(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 56

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 57

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 60

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 62

Given:

```
Integer[] intArray = {2, 1, 3, 4, 5};  
List<Integer> list =  
new ArrayList<>(Arrays.asList (intArray));  
list.parallelStream()  
    .forEach(e -> System.out.print(e + " "));
```

Which two are correct? (Choose two.)

- A. The output will be exactly 2 1 3 4 5.
- B. The program prints 1 4 2 3, but the order is unpredictable.
- C. Replacing forEach() with forEachOrdered(), the program prints 2 1 3 4 5, but the order is unpredictable.
- D. Replacing forEach() with forEachOrdered(), the program prints 1 2 3 4 5.
- E. Replacing forEach() with forEachOrdered(), the program prints 2 1 3 4 5.

**Answer: BD**

**Explanation:**



```

8 public class Secret {
9     public static void main(String[] args) {
10         Integer[] intArray = {1, 2, 3, 4, 5};
11         List<Integer> list =
12             new ArrayList<> (Arrays.asList (intArray));
13         list.parallelStream()
14             .forEachOrdered(e -> System.out.print(e + " "));
15     }
16 }

```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.32 sec(s), Memory: 37040 kilobyte(s)

1 2 3 4 5

#### NEW QUESTION 63

Examine these module declarations:

```

module ServiceAPI {
    exports com.example.api;
}

module ServiceProvider {
    requires ServiceAPI;
    provides com.example.api with com.myimpl.Impl;
}

module Consumer {
    requires ServiceAPI;
    uses com.example.api;
}

```

Which two statements are correct? (Choose two.)

- A. The ServiceProvider module is the only module that, at run time, can provide the com.example.api API.
- B. The placement of the com.example.api API in a separate module, ServiceAPI, makes it easy to install multiple provider modules.
- C. The Consumer module should require the ServiceProvider module.
- D. The ServiceProvider module should export the com.myimpl package.
- E. The ServiceProvider module does not know the identity of a module (such as Consumer) that uses the com.example.api API.

**Answer:** AC

#### NEW QUESTION 68

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 69

Given:

```
public class Main {
    public static void main(String[] args) {
        int i = 1;
        for(String s : args) {
            System.out.println((i++) + " " + s);
        }
    }
}
```

executed with this command: java Main one two three

What is the output of this class?

- A. The compilation fails.
- B. 1) one2) two3) three
- C. A java.lang.ArrayIndexOutOfBoundsException is thrown.
- D. 1) one
- E. nothing

**Answer: B**

#### NEW QUESTION 72

Given:

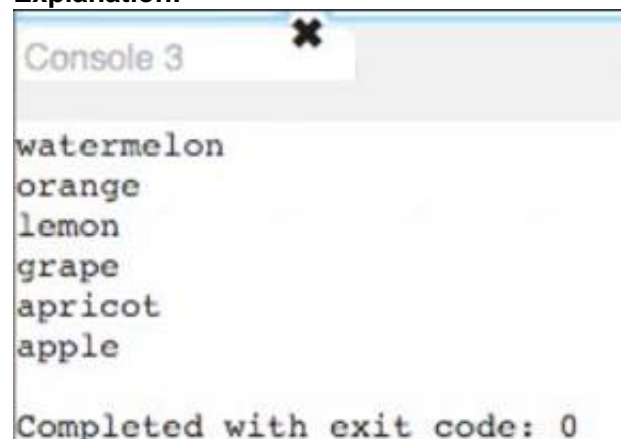
```
import java.util.ArrayList;
import java.util.Arrays;
public class NewMain {
    public static void main(String[] args) {
        String[] fruitNames = { "apple", "orange",
            "grape", "lemon", "apricot", "watermelon" };
        var fruits = new ArrayList<>(Arrays.asList(fruitNames));
        fruits.sort((var a, var b) -> -a.compareTo(b));
        fruits.forEach(System.out::println);
    }
}
```

What is the result?

- A. watermelonorangelemongrapeapricotapple
- B. nothing
- C. appleapricotgrapelemonorangewatermelon
- D. appleorangegrapelemonapricotwatermelon

**Answer: A**

**Explanation:**



```
Console 3
watermelon
orange
lemon
grape
apricot
apple
Completed with exit code: 0
```

#### NEW QUESTION 74

Which three initialization statements are correct? (Choose three.)

- A. int x = 12\_34;
- B. short sh = (short)'A';
- C. String contact# = "(+2) (999) (232)";

- D. boolean true = (4 == 4);
- E. float x = 1.99;
- F. int[][] e = {{1,1},{2,2}};
- G. byte b = 10;char c = b;

**Answer:** ABF

#### NEW QUESTION 79

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 83

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 86

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();
    }
}

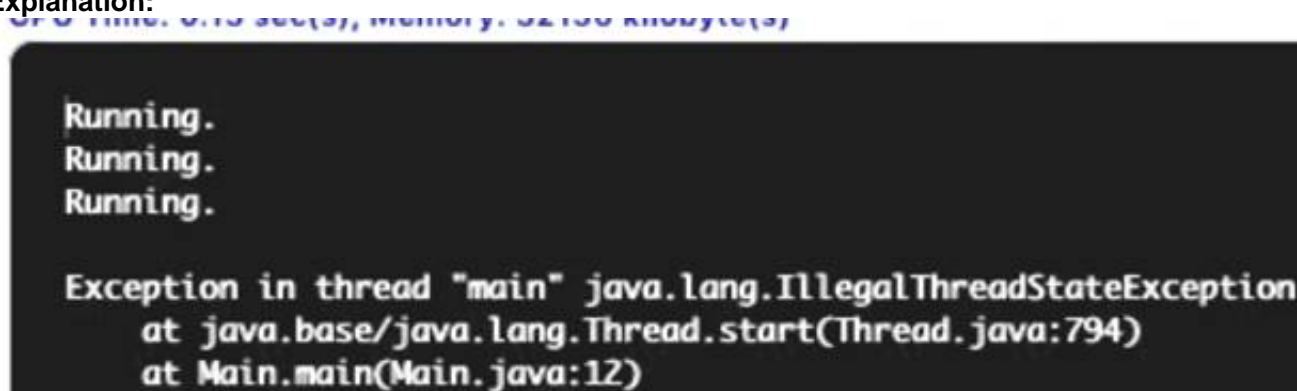
class MyThread implements Runnable {
    public void run() {
        System.out.println("Running.");
    }
}
```

Which one is correct?

- A. An `IllegalThreadStateException` 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 91

Given the code fragment:

```
String s = "";
if (Double.parseDouble("11.00f") > 11) {
    s += 1;
}
if (1_7 == Integer.valueOf("17")) {
    s += 2;
}
if (1024 > 1023L) {
    s += 3;
}
System.out.print(s);
```

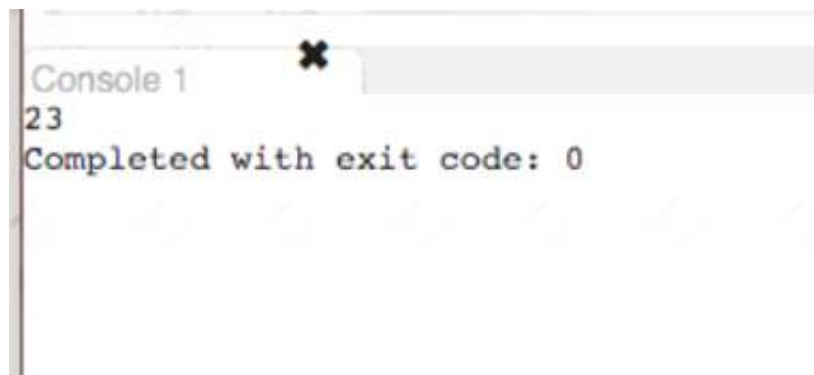
What is the result?

- A. 23
- B. 12
- C. 123
- D. 13

**Answer:** A

**Explanation:**





```

Console 1
23
Completed with exit code: 0
  
```

#### NEW QUESTION 96

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
Completed with exit code: 0
  
```

#### NEW QUESTION 100

Which statement about access modifiers is correct?

- A. An instance variable can be declared with the static modifier.
- B. A local variable can be declared with the final modifier.
- C. An abstract method can be declared with the private modifier.
- D. An inner class cannot be declared with the public modifier.
- E. An interface can be declared with the protected modifier.

**Answer: B**

**NEW QUESTION 102**

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 105**

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 106

Given the Person class with age and name along with getter and setter methods, and this code fragment:

```

List<Person> persons = new ArrayList(List.of(new Person(44,"Tom"),
                                              new Person(40,"Aman"),
                                              new Person(40,"Peter")));

persons.sort(Comparator.comparing((Person::getAge))
                  .thenComparing(Person::getName)
                  .reversed());

persons.forEach(p1->System.out.print(" "+p1.getName()));

```

What will be the result?

- A. Aman Tom Peter
- B. Tom Aman Peter
- C. Aman Peter Tom
- D. Tom Peter Aman

**Answer: C**

## NEW QUESTION 111

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 116

Given:

```

public class Employee {
    private String name;
    private String locality;
    /* the constructor, getter and setter methods code goes here */
}

```

and:

```

8. List<Employee> roster = new ArrayList<>();
9. long empCount = roster.stream()
10. /* insert code here */
11. System.out.print(empCount);

```

Which code, when inserted on line 10, prints the number of unique localities from the roster list?

- A. `.map(Employee::getLocality).distinct().count();`
- B. `map(e -> e.getLocality()).count();`
- C. `.map(e -> e.getLocality()).collect(Collectors.toSet()).count();`
- D. `.filter(Employee::getLocality).distinct().count();`

**Answer: D**

#### NEW QUESTION 119

Given:

```

public interface ExampleInterface{ }

```

Which two statements are valid to be written in this interface? (Choose two.)

- A. `public abstract void methodB();`
- B. `final void methodG(){System.out.println("G");}`
- C. `private abstract void methodC();`
- D. `public String methodD();`
- E. `public int x;`
- F. `final void methodE();`
- G. `public void methodF(){System.out.println("F");}`

**Answer: AD**

#### NEW QUESTION 123

Given:

```

public class Main {
    public static void main(String[] args) {
        List l = new ArrayList();
        l.add("hello");
        l.add("world");
        print(l);
    }
    private static void print(List<String>... args) {
        for (List<String> str : args) {
            System.out.println (str);
        }
    }
}

```

Which annotation should be used to remove warnings from compilation?

- A. `@SuppressWarnings` on the main and print methods
- B. `@SuppressWarnings("unchecked")` on main and `@SafeVarargs` on the print method
- C. `@SuppressWarnings("rawtypes")` on main and `@SafeVarargs` on the print method
- D. `@SuppressWarnings("all")` on the main and print methods



Answer: B

Explanation:

```
13 @SuppressWarnings("unchecked")
14 public class Main {
15
16     public static void main(String[] args) {
17
18         List l = new ArrayList();
19         l.add("Hello");
20         l.add("world");
21         print(l);
22
23     }
24
25     private static void print(List<String>... args) {
26         for (List<String> str : args) {
27             System.out.println (str);
28         }
29     }
30 }
31 @SafeVarargs
32 }
```

#### NEW QUESTION 125

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 126

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:**



Completed with exit code: 0

#### NEW QUESTION 127

Given:

List<String> longlist = List.of("Hello", "World", "Beat"); List<String> shortlist = new ArrayList<>();

Which code fragment correctly forms a short list of words containing the letter "e"?

- A. 

```
longList.stream()
    .filter(w -> w.indexOf('e') != -1)
    .parallel()
    .forEach(w -> shortList.add(w));
```
- B. 

```
longList.parallelStream()
    .filter(w -> w.indexOf('e') != -1)
    .forEach(w -> shortList.add(w));
```
- C. 

```
shortList = longList.stream()
    .filter(w -> w.indexOf('e') != -1)
    .parallel()
    .collect(Collectors.toList());
```
- D. 

```
longList.stream()
    .filter(w -> w.indexOf('e') != -1)
    .parallel()
    .collect(shortlist);
```

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

**Answer: C**

#### NEW QUESTION 130

Given:

```
public class Over {  
    public void analyze(Object[] o){  
        System.out.println("I am an object array");  
    }  
    public void analyze(long[] l){  
        System.out.println("I am an array");  
    }  
    public void analyze(Object o){  
        System.out.println("I am an object");  
    }  
    public static void main(String[] args) {  
        int[] nums = new int[10];  
        new Over().analyze(nums);    // line 1  
    }  
}
```

What is the output?

- A. I am an object array
- B. The compilation fails due to an error in line 1.
- C. I am an array
- D. I am an object

**Answer:** D

#### NEW QUESTION 135

Given:

```
List<String> list1 = new ArrayList<>(); list1.add("A");  
list1.add("B");  
List list2 = List.copyOf(list1); list2.add("C");  
List<List<String>> list3 = List.of(list1, list2); System.out.println(list3);  
What is the result?
```

- A. [[A, B],[A, B]]
- B. An exception is thrown at run tim
- C. [[A, B], [A, B, C]]
- D. [[A, B, C], [A, B, C]]

**Answer:** B

**Explanation:**

```

11
12 public class Main {
13     public static void main(String[] args) {
14
15         List<String> list1 = new ArrayList<>();
16         list1.add("A");
17         list1.add("B");
18         List list2 = List.copyOf(list1);
19         list2.add("C");
20         List<List<String>> list3 = List.of(list1, list2);
21         System.out.println(list3);
22     }
23
24 }
25

```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4



Interactive

Stdin Inputs

CommandLine Arguments

Execute



Result

CPU Time: 0.16 sec(s), Memory: 32128 kilobyte(s)

```

Exception in thread "main" java.lang.UnsupportedOperationException
    at java.base/java.util.ImmutableCollections.uoe(ImmutableCollections.java:71)
    at java.base/java.util.ImmutableCollections$AbstractImmutableCollection.add(ImmutableCollections.java:75)
    at Main.main(Main.java:19)

```

**NEW QUESTION 137**

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 139**

Given:



```
class CustomType<T> {
    public <T> int count(T[] anArray, T element) {
        int count = 0;
        for(T e : anArray) {
            if (e.equals(element)) ++count;
        }
        return count;
    }
}

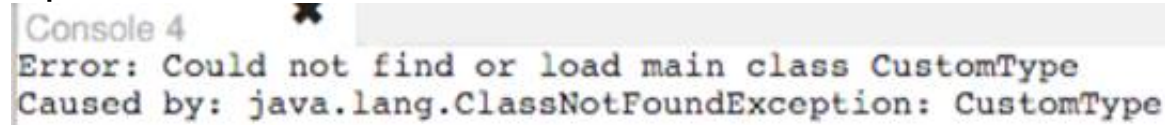
and
public class Test extends CustomType {
    public static void main(String[] args) {
        String[] words = {"banana","orange","apple","lemon"};
        Integer[] numbers = {1,2,3,4,5};
        CustomType type = new CustomType();
        CustomType<String> stringType = new CustomType<>();
        System.out.println(stringType.count(words, "apple"));
        System.out.println(type.count(words, "apple"));
        System.out.println(type.count (numbers, 3));
    }
}
```

What is the result?

- A. A NullPointerException is thrown at run time.
- B. The compilation fails.
- C. 1Null null
- D. 111
- E. A ClassCastException is thrown at run time.

**Answer: B**

**Explanation:**



Console 4  
Error: Could not find or load main class CustomType  
Caused by: java.lang.ClassNotFoundException: CustomType

#### NEW QUESTION 144

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 149

.....

## Thank You for Trying Our Product

### We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questions and Answers in PDF Format

### 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 First Try
- \* 1Z0-819 Practice Test Questions in Multiple Choice Formats and Updates for 1 Year

**100% Actual & Verified — Instant Download, Please Click**  
**[Order The 1Z0-819 Practice Test Here](#)**