

1Z0-819 Dumps

Java SE 11 Developer

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



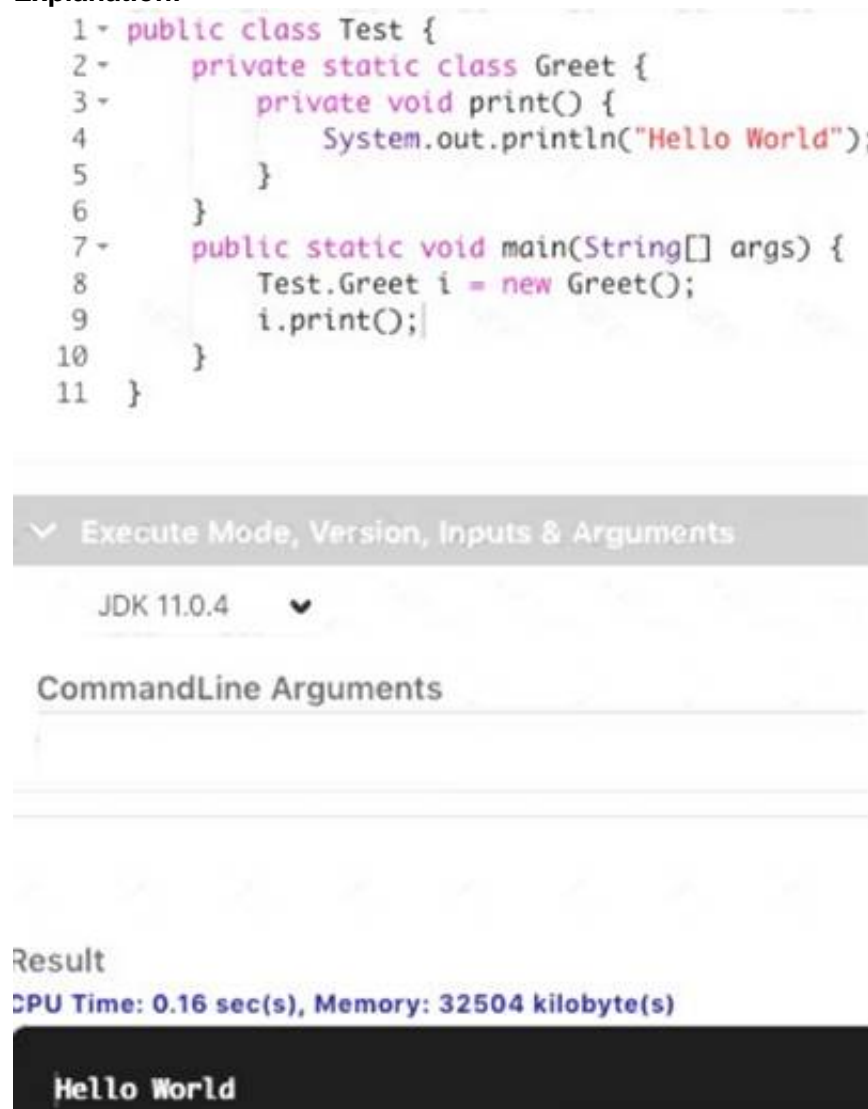
NEW QUESTION 1

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:

```
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

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 4

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 5

Which interface in the java.util.function package will return a void return type?

- A. Supplier

- B. Predicate
- C. Function
- D. Consumer

Answer: D

NEW QUESTION 6

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 7

Which two commands are used to identify class and module dependencies? (Choose two.)

- A. jmod describe
- B. java Hello.java
- C. jdeps --list-deps
- D. jar --show-module-resolution
- E. java --show-module-resolution

Answer: CE

NEW QUESTION 8

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 9

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 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

Given:

```

public class Foo {
    public <T> Collection<T> foo(Collection<T> arg) { ... }
}

```

and

```

public class Bar extends Foo { ... }

```

Which two statements are true if the method is added to Bar? (Choose two.)

- A. public Collection<String> foo(Collection<String> arg) { ... } overrides Foo.foo.
- B. public <T> Collection<T> foo(Stream<T> arg) { ... } overloads Foo.foo.
- C. public <T> List<T> foo(Collection<T> arg) { ... } overrides Foo.foo.
- D. public <T> Collection<T> foo(Collection<T> arg) { ... } overloads Foo.foo.
- E. public <T> Collection<T> bar(Collection<T> arg) { ... } overloads Foo.foo.
- F. public <T> Iterable<T> foo(Collection<T> arg) { ... } overrides Foo.foo.

Answer: CF

NEW QUESTION 13

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 16

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 17

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 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 24

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 28

Given:

```
public class Main {

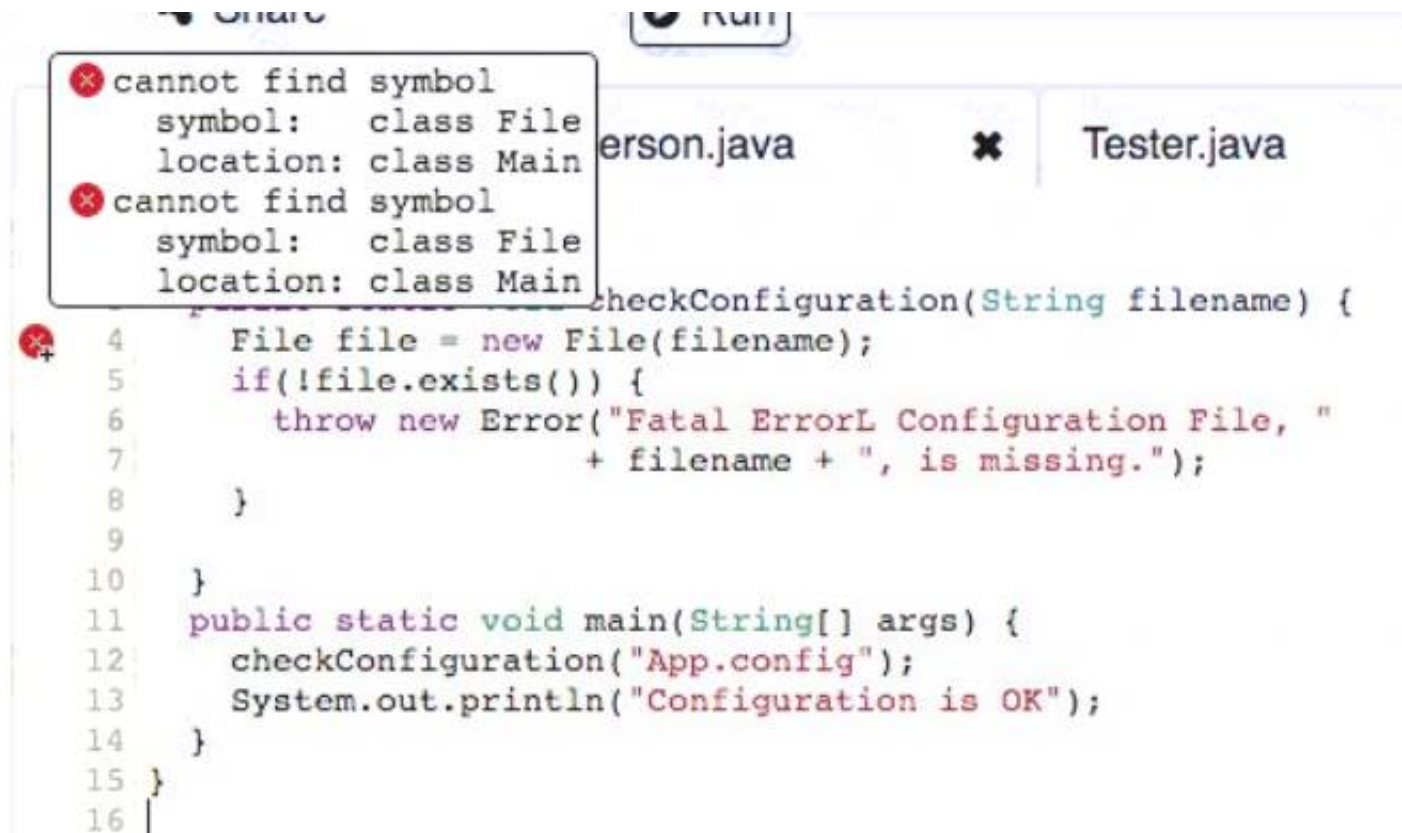
    public static void checkConfiguration(String filename) {
        File file = new File(filename);
        if(!file.exists()) {
            throw new Error("Fatal Error: Configuration File, "
                + filename + ", is missing.");
        }
    }

    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 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 34

Given:

```

public class Tester {
    public static void main(String[] args) {
        int x = 4;
        int y = 2;
        System.out.println(x+y+"=(x+y)="+x+y);
    }
}

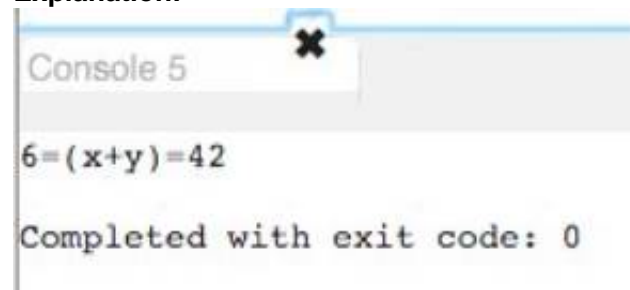
```

What is the result?

- A. An exception is thrown at runtime
- B. 42=(x+y)=42
- C. 42=(x+y)=6
- D. 6=(x+y)=42
- E. 6=(x+y)=6

Answer: D

Explanation:



NEW QUESTION 38

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 42

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 46

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 51

What makes Java dynamic?

- A. At runtime, classes are loaded as needed, and new code modules can be loaded on demand.
- B. The runtime can process machine language sources as well as executables from different language compilers.
- C. The Java compiler uses reflection to test if class methods are supported by resources of a target platform.
- D. The Java compiler preprocesses classes to run on specific target platforms.

Answer: A

NEW QUESTION 54

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 57

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 62

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 65

Given:

```
public class Person {
    private String name;
    public void setName(String name) {
        String title = "Dr. ";
        name = title+name;
    }
    public String toString() {
        return name;
    }
}
```

and

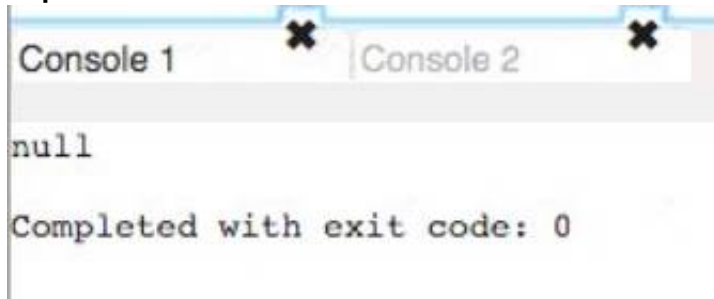
```
public class Test {
    public static void main(String args[]) {
        Person p = new Person();
        p.setName("Who");
        System.out.println(p);
    }
}
```

What is the result?

- A. D
- B. Who
- C. D
- D. Null
- E. An exception is thrown at runtime.
- F. null

Answer: D

Explanation:



NEW QUESTION 68

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 70

Given:

```
public class Foo {  
    private final ReentrantLock lock = new ReentrantLock();  
    private State state;  
    public void foo() throws Exception {  
        try {  
            lock.lock();  
            state.mutate();  
        }  
        finally {  
            lock.unlock();  
        }  
    }  
}
```

What is required to make the Foo class thread safe?

- A. No change is required.
- B. Make the declaration of lock static.
- C. Replace the lock constructor call with new ReentrantLock (true).
- D. Move the declaration of lock inside the foo method.

Answer: C

NEW QUESTION 71

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 76

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 77

Given the code fragment:

```
String s1 = new String("ORACLE");
String s2 = "ORACLE";
String s3 = s1.intern();
```

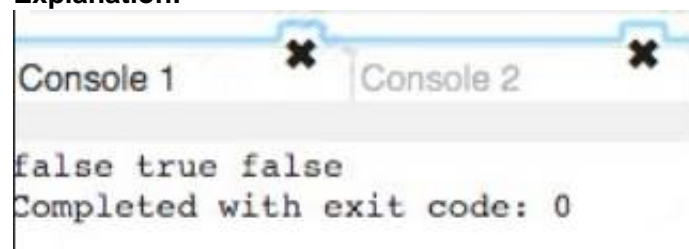
```
System.out.print((s1==s2) + " ");
System.out.print((s2==s3) + " ");
System.out.println(s1==s3);
```

What is the result?

- A. false true true
- B. true false false
- C. false false true
- D. false true false

Answer: D

Explanation:



NEW QUESTION 81

Given:

```
@Target(ElementType.METHOD)
@Retention(RetentionPolicy.RUNTIME)
public @interface AuthorInfo {
    String author() default "";
    String date();
    String[] comments() default {};
}
```

Which two are correct? (Choose two.)

- A. `@AuthorInfo(date="1-1-2020", comments={ null })`
`public class Hello {`
 `public void func() {}`
`}`
- B. `public class Hello {`
 `@AuthorInfo (date="1-1-2020. comments="Hello")`
 `public void func() {}`
`}`
- C. `public class Hello {`
 `@AuthorInfo`
 `public void func() {}`
`}`
- D. `@AuthorInfo(date="1-1-2020")`
`public class Hello {`
 `public void func() {}`
`}`
- E. `public class Hello {`
 `@AuthorInfo(date="1-1-2020", author="Gandhi", comments={ "world" })`
 `public void func () {}`
`}`

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

Answer: CD

NEW QUESTION 86

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 87

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 92

Given:

```

class Employee {
    String office;
}

```

and the code fragment:

```

5. public class HRApp {
6.     var employee = new ArrayList<Employee>();
7.     public var display() {
8.         var employee = new Employee();
9.         var offices = new ArrayList<>();
10.        offices.add("Chicago");
11.        offices.add("Bangalore");
12.        for (var office : offices) {
13.            System.out.print("Employee Location"+ office);
14.        }
15.    }
16. }

```

Which two lines cause compilation errors? (Choose two.)

- A. line 12
- B. line 6
- C. line 9
- D. line 8
- E. line 7

Answer: BE

NEW QUESTION 94

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 95

You are working on a functional bug in a tool used by your development organization. In your investigation, you find that the tool is executed with a security policy file containing this grant.

```
grant codebase "file:${klib.home}/j2se/home/klib.jar" {  
    permission java.security.AllPermission;  
};
```

What action should you take?

A. Nothing, because it is an internal tool and not exposed to the public.

B. Remove the grant because it is excessive.

C. Nothing, because it is not related to the bug you are investigating.

D. File a security bug against the tool referencing the excessive permission granted.

E. Nothing, because listing just the required permissions would be an ongoing maintenance challenge.

Answer: D

NEW QUESTION 96

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 98

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 99

Given:

```
public class MyResource {
    public MyResource () {
    }
    // Resource methods
}
```

You want to use the myResource class in a try-with-resources statement. Which change will accomplish this?

- A. Extend AutoCloseable and override the close method.
- B. Implement AutoCloseable and override the autoClose method.
- C. Extend AutoCloseable and override the autoClose method.
- D. Implement AutoCloseable and override the close method.

Answer: D

NEW QUESTION 102

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 105

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 110

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 112

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 113

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 117

Given:

```
public interface TestInterface {
    default void samplingProbeProcedure() {
        probeProcedure();
        System.out.println("Collect Sample");
        System.out.println("Leave Asteroid");
        System.out.println("Dock with Main Craft");
    }
    default void explosionProbeProcedure() {
        probeProcedure();
        System.out.println("Explode")
    }
}
```

Examine these requirements:

- Eliminate code duplication.
- Keep constant the number of methods other classes may implement from this interface. Which method can be added to meet these requirements?

- A. `private default void probeProcedure(){
 System.out.println("Launch Probe");
 System.out.println("Land on Asteroid");
}`
- B. `static void probeProcedure(){
 System.out.println("Launch Probe");
 System.out.println("Land on Asteroid");
}`
- C. `private void probeProcedure(){
 System.out.println("Launch Probe");
 System.out.println("Land on Asteroid");
}`
- D. `default void probeProcedure(){
 System.out.println("Launch Probe");
 System.out.println("Land on Asteroid");
}`

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

Answer: B

NEW QUESTION 118

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 120

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 125

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 130

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 134

Given:

```
try {  
    // line 1  
    lines.map(l -> l.toUpperCase())  
        .forEach (line --> {  
            try {  
                Files.write(Paths.get("outputFile_to_path"),  
line.getBytes(), StandardOpenOption.CREATE);  
            } catch (IOException e) {  
                e.printStackTrace();  
            }  
        });  
} catch (IOException e) {  
    e.printStackTrace();  
}
```

You want to obtain the Stream object on reading the file. Which code inserted on line 1 will accomplish this?

- A. var lines = Files.lines(Paths.get(INPUT_FILE_NAME));
B. Stream lines = Files.readAllLines(Paths.get(INPUT_FILE_NAME));
C. var lines = Files.readAllLines(Paths.get(INPUT_FILE_NAME));
D. Stream<String> lines = Files.lines(INPUT_FILE_NAME);

Answer: C

NEW QUESTION 137

Given:

```
public class Test {  
    public static void main(String[] args) {  
        int x;  
        int y = 5;  
        if (y > 2) {  
            x = ++y;  
            y = x + 7;  
        } else {  
            y++;  
        }  
        System.out.print(x + " " + y);  
    }  
}
```

What is the result?

- A. compilation error
B. 0 5
C. 6 13
D. 5 12

Answer: A

Explanation:

```

1 public class Test {
2     public static void main (String[] args) {
3         int x;
4         int y = 5;
5         if (y > 2) {
6             x = ++y;
7             y = x + 7;
8         } else {
9             y++;
11        System.out.print(x + " "+y);
12    }
13 }

```

variable x might not have been initialized

NEW QUESTION 138

Given:

```

public class Confidential implements Serializable{
    private String data;

    public Confidential(String data) {
        this.data = data;
    }
}

```

Which two are secure serialization of these objects? (Choose two.)

- A. Define the serialPersistentFields array field.
- B. Declare fields transient.
- C. Implement only readResolve to replace the instance with a serial proxy and not writeReplace.
- D. Make the class abstract.
- E. Implement only writeReplace to replace the instance with a serial proxy and not readResolve.

Answer: AC

NEW QUESTION 141

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 142

Which two are functional interfaces? (Choose two.)

- A. `@FunctionalInterface`

```

interface MyRunnable {
    public void run();
}

```
- B. `@FunctionalInterface`

```

interface MyRunnable {
    public void run();
    public void call();
}

```
- C. `interface MyRunnable {`

```

    public default void run() {}
    public void run(String s);
}

```
- D. `@FunctionalInterface`

```

interface MyRunnable {
}

```
- E. `interface MyRunnable {`

```

    @FunctionalInterface
    public void run();
}

```

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

Answer: CE

NEW QUESTION 143

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 146

Given:

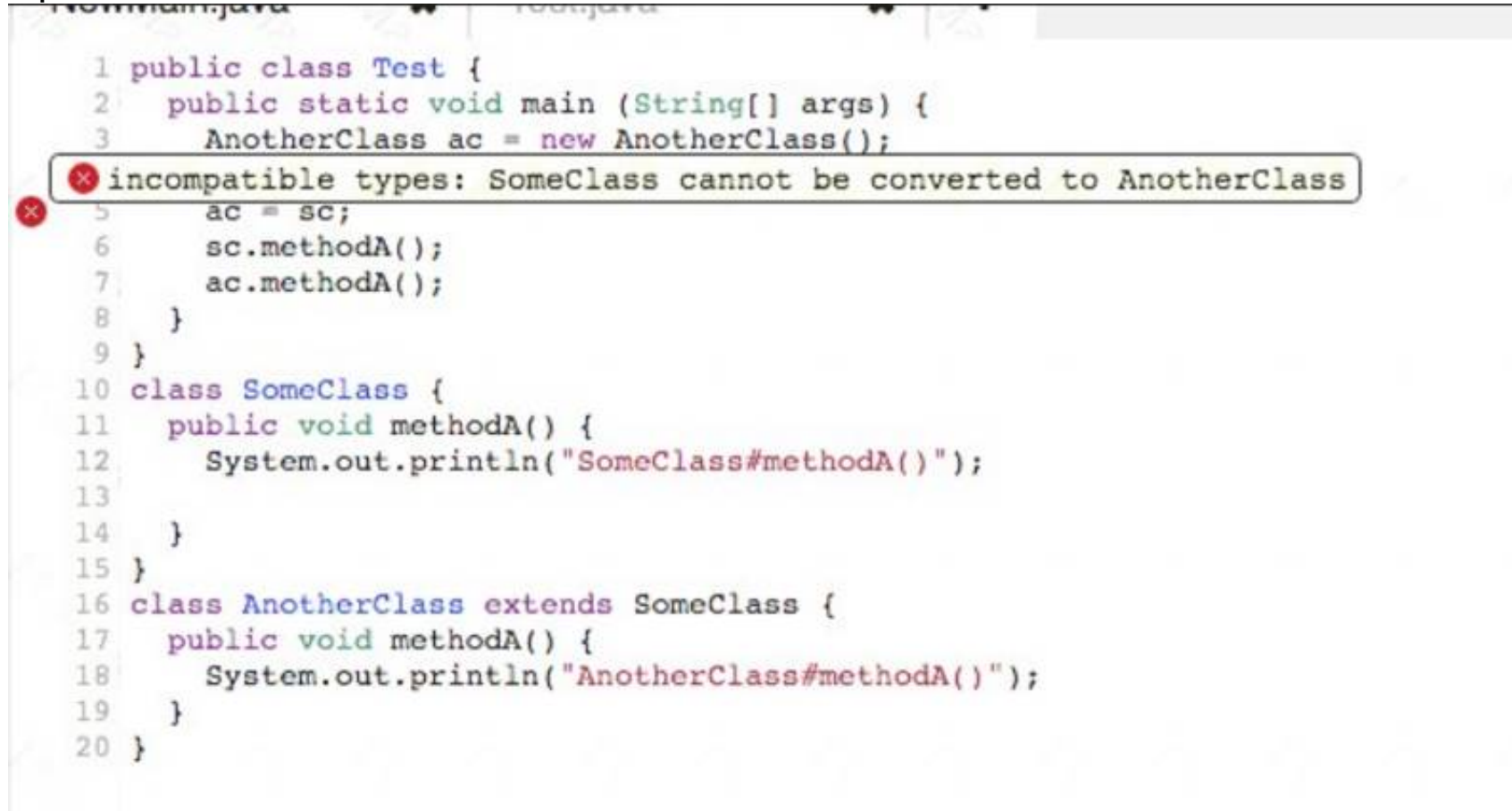
```
public class Test {
    public static void main(String[] args) {
        AnotherClass ac = new AnotherClass();
        SomeClass sc = new AnotherClass();
        ac = sc;
        sc.methodA();
        ac.methodA();
    }
}
class SomeClass {
    public void methodA() {
        System.out.println("SomeClass#methodA()");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



NEW QUESTION 150

Which set of commands is necessary to create and run a custom runtime image from Java source files?

- A. java, jdeps
- B. javac, jlink
- C. jar, jlink
- D. javac, jar

Answer: B

NEW QUESTION 153

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 157

Given:

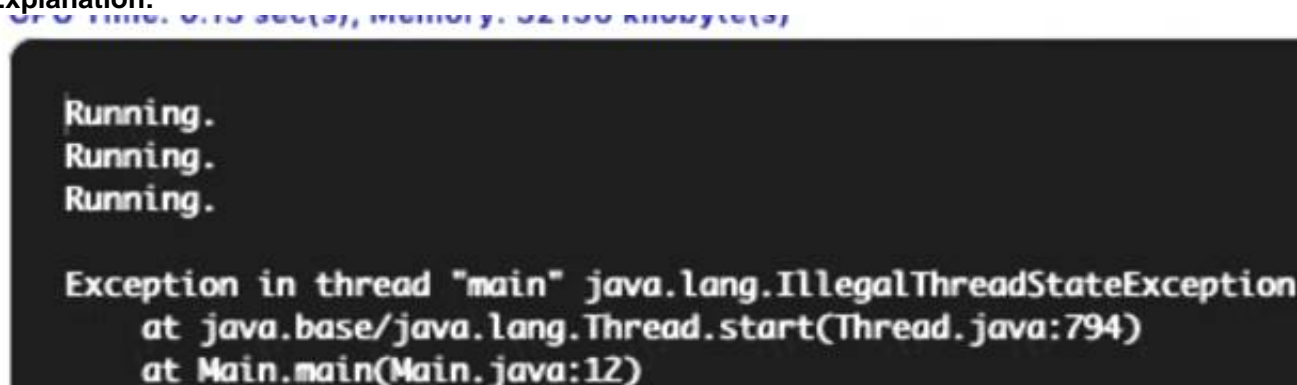
```
public class Main {  
    public static void main(String[] args) {  
        Thread t1 = new Thread(new MyThread());  
        Thread t2 = new Thread(new MyThread());  
        Thread t3 = new Thread(new MyThread());  
  
        t1.start();  
        t2.run();  
        t3.start();  
  
        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 161

Given the code fragment:

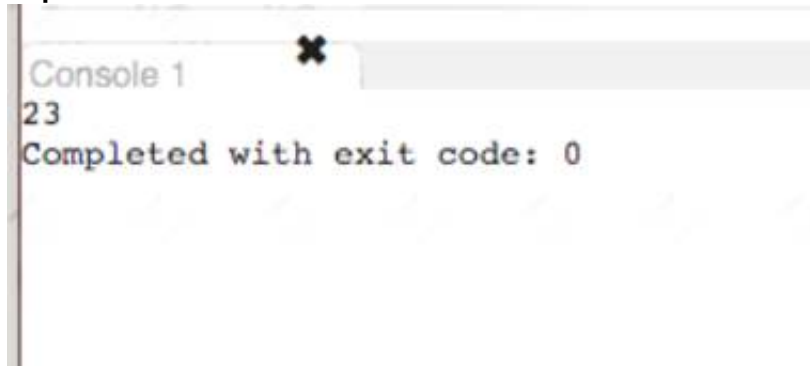
```
String s = "";
if (Double.parseDouble("11.00f") > 11) {
    s += 1;
}
if (1_7 == Integer.valueOf("17")) {
    s += 2;
}
if (1024 > 1023L) {
    s += 3;
}
System.out.print(s);
```

What is the result?

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

Answer: A

Explanation:



NEW QUESTION 163

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);
    }
}

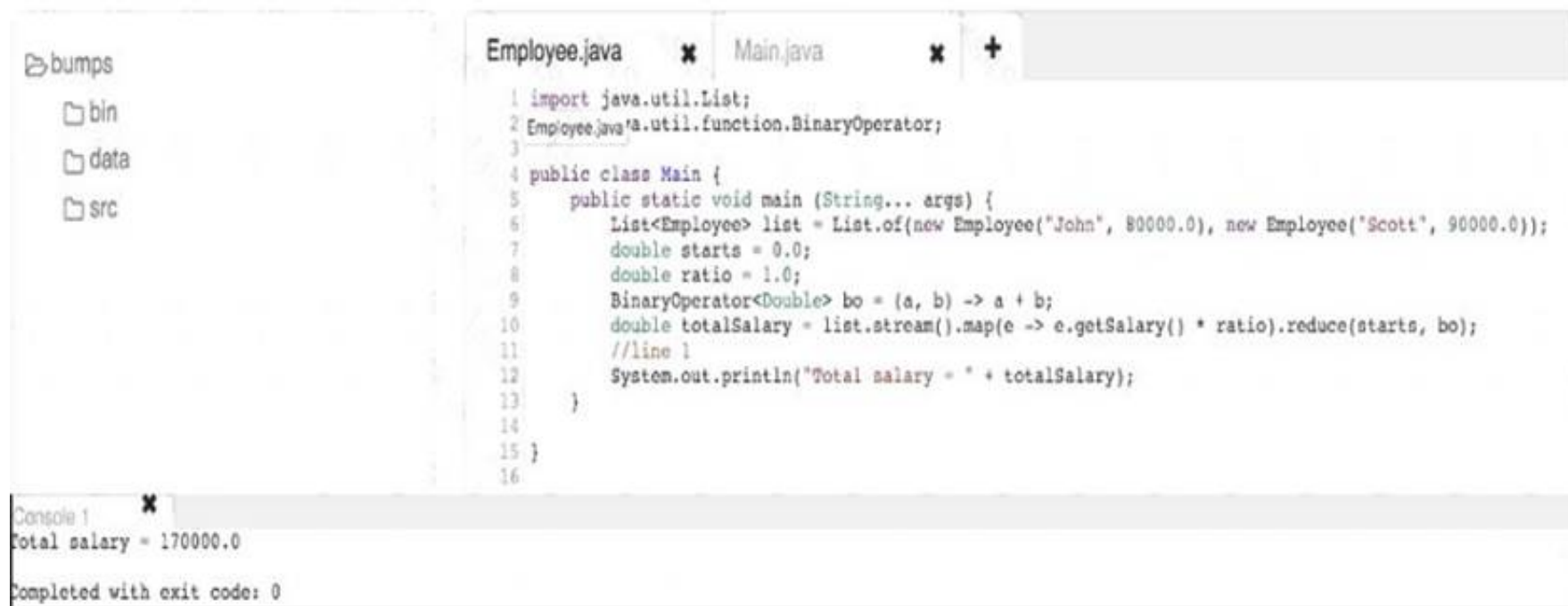
class Employee {
    String name;
    double salary;
    public Employee(String name, double salary) {
        this.name = name;
        this.salary = salary;
    }
    public String getName() { return name; }
    public double getSalary() { return salary; }
}
```

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 following code:

```
1 import java.util.List;
2 import java.util.function.BinaryOperator;
3
4 public class Main {
5     public static void main (String... args) {
6         List<Employee> list = List.of(new Employee("John", 80000.0), new Employee("Scott", 90000.0));
7         double starts = 0.0;
8         double ratio = 1.0;
9         BinaryOperator<Double> bo = (a, b) -> a + b;
10        double totalSalary = list.stream().map(e -> e.getSalary() * ratio).reduce(starts, bo);
11        //line 1
12        System.out.println("Total salary = " + totalSalary);
13    }
14 }
15 }
16 }
```

Below the code editor, the console output is visible:

```
Console 1
Total salary = 170000.0
Completed with exit code: 0
```

NEW QUESTION 167

Given:

```
interface MyInterface1 {
    public int method() throws Exception;
    private void pMethod() { /* an implementation of pMethod */ }
}

interface MyInterface2 {
    public static void sMethod() { /* an implementation of sMethod */ }
    public boolean equals();
}

interface MyInterface3 {
    public void method();
    public void method(String str);
}

interface MyInterface4 {
    public void dMethod() { /* an implementation of dMethod */ }
    public void method();
}

interface MyInterface5 {
    public static void sMethod();
    public void method(String str);
}
```

Which two interfaces can be used in lambda expressions? (Choose two.)

- A. MyInterface1
- B. MyInterface3
- C. MyInterface5
- D. MyInterface2
- E. MyInterface4

Answer: CD

NEW QUESTION 171

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 175

Given:

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

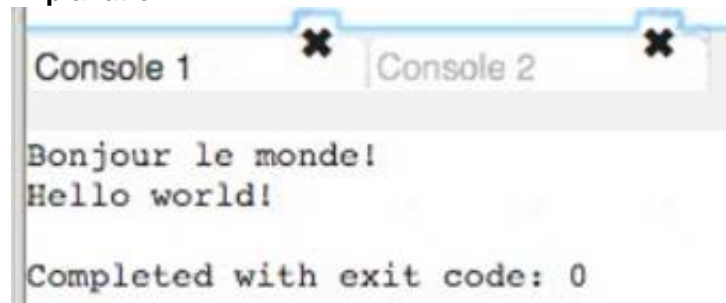
public class Bar extends Foo {
    private void print() {
        System.out.println("Hello world!");
    }
    public void bar() {
        print();
    }
    public static void main(String... args) {
        Bar b = new Bar();
        b.foo();
        b.bar();
    }
}
```

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 178

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 183

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 184

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 189

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 193

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 196

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 198

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 199

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 203**

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 204

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 205

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 209

Given:

```
public interface InterfaceOne {
    void printOne();
}
```

Which three classes successfully override printOne()? (Choose three.)

- A.
- ```
public abstract class TestClass implements InterfaceOne {
 public abstract void printOne();
}
```
- B.
- ```
public class TestClass implements InterfaceOne {  
    private void printOne(){  
        System.out.println("one");  
    }  
}
```
- C.
- ```
public class TestClass implements InterfaceOne {
 public void printOne(){
 System.out.println("one");
 }
}
```
- D.
- ```
public abstract class TestClass implements InterfaceOne {  
    public void printOne(){  
        System.out.println("one");  
    }  
}
```
- E.
- ```
public abstract class TestClass implements InterfaceOne {
 public String printOne(){
 return "one";
 }
}
```
- F.
- ```
public class TestClass{  
    public void printOne(){  
        System.out.println("one");  
    }  
}
```

- A. Option A
B. Option B
C. Option C
D. Option D
E. Option E
F. Option F

Answer: ACD

NEW QUESTION 212

Given:

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

    void setACount(int cCount){
        cCount = cCount;
    }
    void setTCount(){
        this.tCount = tCount;
    }
    int setCCount(){
        return cCount;
    }
    int setGCount(int g){
        gCount = g;
        return gCount;
    }
    void setAllCounts(int x){
        aCount = tCount = this.cCount = setGCount(x);
    }
}
```

Which two methods modify field values? (Choose two.)

- A. setAllCounts
- B. setACount
- C. setGCount
- D. setCCount
- E. setTCount

Answer: AC

NEW QUESTION 214

Which code fragment compiles?

- A.

```
Comparator comparator = new Comparator<?>() {
    public int compare(Integer i, Integer j) {
        return i.compareTo(j);
    }
};
```
- B.

```
var comparator = new Comparator<>() {
    public int compare(Integer i, Integer j) {
        return i.compareTo(j);
    }
};
```
- C.

```
Comparator<> comparator = new Comparator<Integer>() {
    public int compare(Integer i, Integer j) {
        return i.compareTo(j);
    }
};
```
- D.

```
Comparator<Integer> comparator = new Comparator<>() {
    public int compare(Integer i, Integer j) {
        return i.compareTo(j);
    }
};
```

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

Answer: D

Explanation:

```

1 import java.io.*;
2 import java.util.*;
3 class abc {
4     public static void main(String[] args) {
5
6         Comparator<Integer> comparator = new Comparator<>() {
7             public int compare(Integer i, Integer j) {
8                 return i.compareTo(j);
9             }
10        };
11    }
12 }
13 }|
14

```

NEW QUESTION 219

Given:

```
StringBuilder s = new StringBuilder("ABCD");
```

Which would cause s to be AQCD?

- A. s.replace(s.indexOf("A"), s.indexOf("C"), "Q");
- B. s.replace(s.indexOf("B"), s.indexOf("C"), "Q");
- C. s.replace(s.indexOf("B"), s.indexOf("B"), "Q");
- D. s.replace(s.indexOf("A"), s.indexOf("B"), "Q");

Answer: B

NEW QUESTION 223

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 226

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 229

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 234

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 238

.....

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>