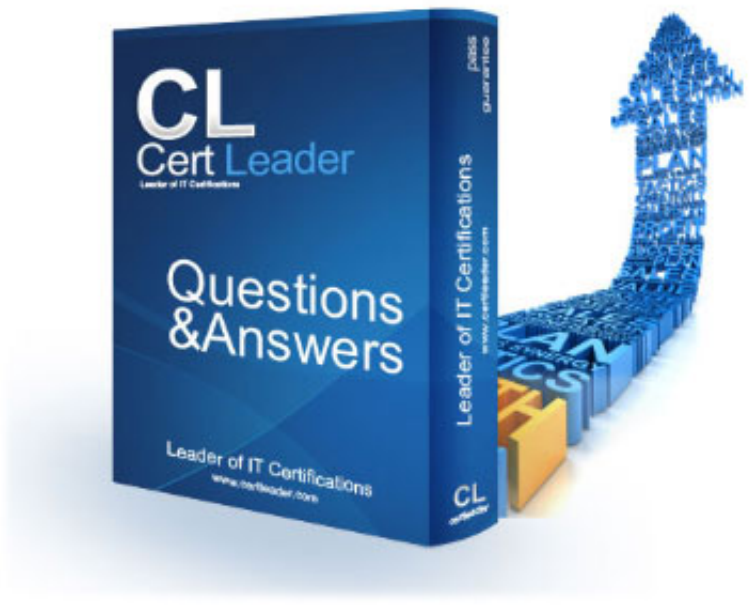
1Z0-819 Dumps

Java SE 11 Developer

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



NEW QUESTION 1
Given:

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

and

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

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

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

Answer: BC

NEW QUESTION 2
Assuming the Widget class has a getPrice method, this code does not compile:

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

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

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

Answer: AD

NEW QUESTION 3
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 4
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 5
Which two statements are true about the modular JDK? (Choose two.)

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

Answer: AC

NEW QUESTION 6
Given:

```
public 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:

NEW QUESTION 7

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 8
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 9
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 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 10
Given the code fragment:
Path currentFile = Paths.get(“/scratch/exam/temp.txt”); Path outputFile = Paths get(“/scratch/exam/new.txt”); Path directory = Paths.get(“/scratch/”);
Files.copy(currentFile, outputFile); Files.copy(outputFile, directory);
Files.delete (outputFile);
The /scratch/exam/temp.txt file exists. The /scratch/exam/new.txt and /scratch/new.txt files do not exist. What is the result?

A. /scratch/exam/new.txt and /scratch/new.txt are deleted.
B. The program throws a FileaAlreadyExistsException.
C. The program throws a NoSuchFileException.
D. A copy of /scratch/exam/new.txt exists in the /scratch directory and /scratch/exam/new.txt is deleted.

Answer: C

Explanation:

```
27  public class Main {
28    public static void main(String[] args) {
29   Path currentFile = Paths.get("/scratch/exam/temp.txt");
30   Path outputFile = Paths.get("/scratch/exam/new.txt");
31   Path directory = Paths.get("/scratch/");
32
33   Files.copy(currentFile, outputFile);
34   Files.copy(outputFile, directory);
35   Files.delete (outputFile);
36  }
37  }
38
```

NEW QUESTION 15
Given:

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

and

```
public class Bar extends Foo {
    public void foo(Collection arg) {
        System.out.println("Hello world!");
    }
    public void foo(List arg) {
        System.out.println("Olá Mundo!");
    }
}
```

and

```
Foo f1 = new Foo();
Foo f2 = new Bar();
Bar b1 = new Bar();
Collection<String> c = new ArrayList<>();
```

Which three are true? (Choose three.)

A. b1.foo(c) prints Bonjour le monde!
B. f1.foo(c) prints Hello world!
C. f1.foo(c) prints Olá Mundo!
D. b1.foo(c) prints Hello world!
E. f2.foo(c) prints Olá Mundo!
F. b1.foo(c) prints Olá Mundo!
G. f2.foo(c) prints Bonjour le monde!
H. f2.foo(c) prints Hello world!
I. f1.foo(c) prints Bonjour le monde!

Answer: BFG

NEW QUESTION 17
Given:
jdeps -jdkinternals C:\workspace4\SimpleSecurity\jar\classes.jar
Which describes the expected output?

A. jdeps lists the module dependencies and the package names of all referenced JDK internal API
B. If any are found, the suggested replacements are output in the console.
C. jdeps outputs an error message that the -jdkinternals option requires either the -summary or the - verbose options to output to the console.
D. The -jdkinternals option analyzes all classes in the .jar and prints all class-level dependencies.
E. The -jdkinternals option analyzes all classes in the .jar for class-level dependencies on JDK internal API
F. If any are found, the results with suggested replacements are output in the console.

Answer: A

Explanation:
-jdkinternals option analyzes all classes in the .jar for class-level dependencies on JDK internal APIs. If any are found, the results with suggested replacements are output in the console.

NEW QUESTION 20
Which two statements set the default locale used for formatting numbers, currency, and percentages? (Choose two.)

A. Locale.setDefault(Locale.Category.FORMAT, “zh-CN”);

B. Locale.setDefault(Locale.Category.FORMAT, Locale.CANADA_FRENCH);
C. Locale.setDefault(Locale.SIMPLIFIED_CHINESE);
D. Locale.setDefault("en_CA");
E. Locale.setDefault("es", Locale.US);

Answer: BD

NEW QUESTION 21
Given:

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

You want the code to produce this output:
John
Joe Jane
Which code fragment should be inserted on line 1 and line 2 to produce the output?

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

Answer: B

NEW QUESTION 26
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 29
Given:

```
public class SerializedMessage implements Serializable {
   String message;
   LocalDateTime createdTime;
   transient LocalDateTime updatedDateTime;;
   SerializedMessage(String message) {
      this.message = message;
      this.createdTime = LocalDateTime.now();
   }
   private void readObject (ObjectInputStream in) {
      try {
         in.defaultReadObject();
         this.updatedDateTime = 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 30
Given:

```
class Mycar {
}
```

and

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

What is the expected result of javac?

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

Answer: B

NEW QUESTION 34
Given the code fragment:
var pool = Executors.newFixedThreadPool(5);
Future outcome = pool.submit(() > 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 37
Given:

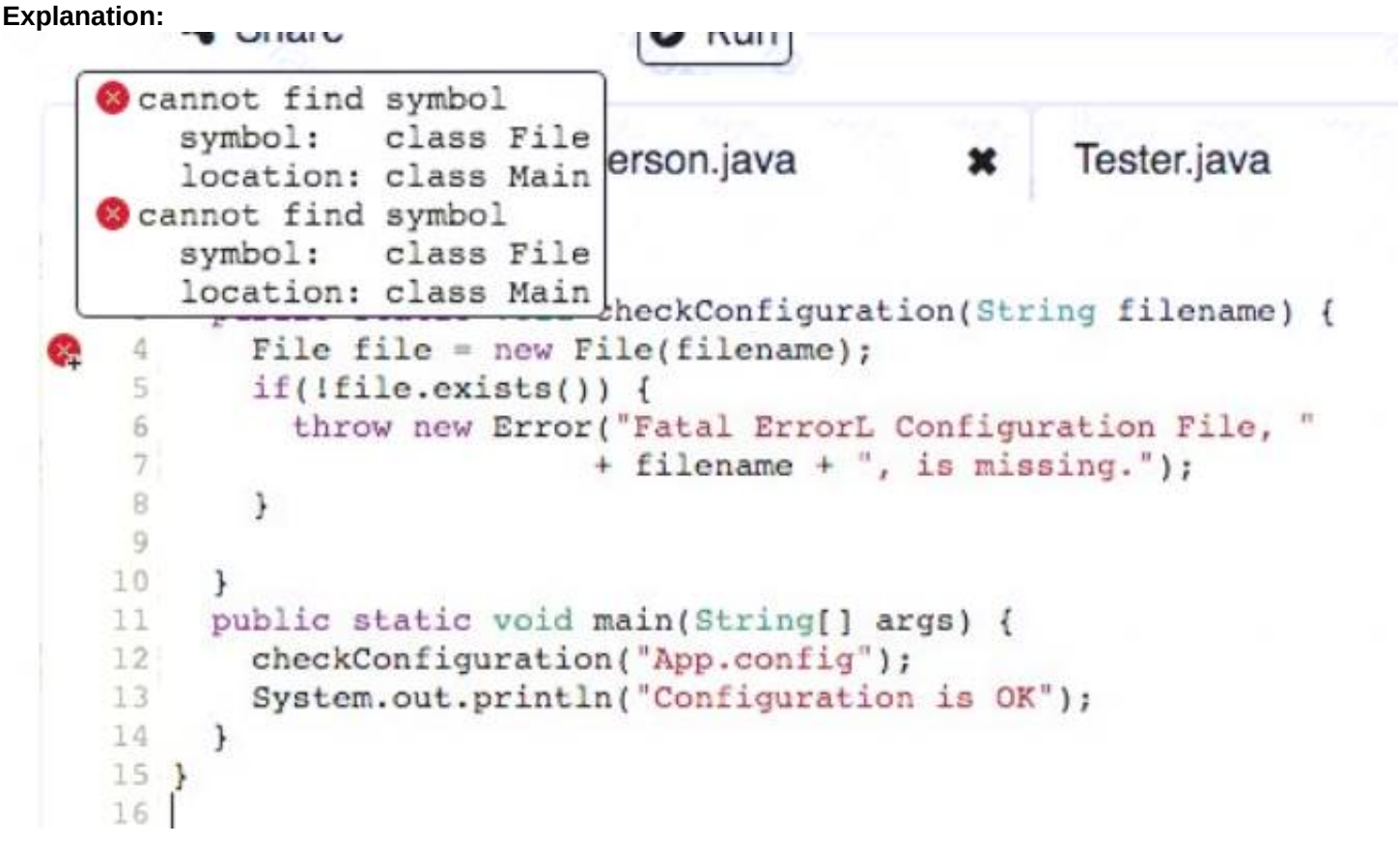
```
public class Main {

  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 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 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 44
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 }
```

NEW QUESTION 45
Given:

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

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

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

C. Add extends Exception on line 1.Add extends Exception on line 2.
D. Add extends Exception on line 1.Add extends GeneralException on line 2.

Answer: D

Explanation:

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

NEW QUESTION 50
Given:

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

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

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

Answer: C

NEW QUESTION 54
Given:

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

and

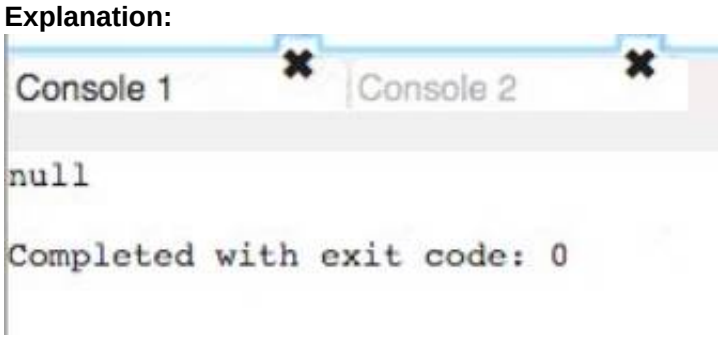
```
public class Test {
   public static void main(String args[]) {
      Person p = new Person();
      p.setName("Who");
      System.out.println(p);
   }
}
```

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 57
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 58
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 63
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 67
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 70
Which is the correct order of possible statements in the structure of a Java class file?

A. class, package, import
B. package, import, class
C. import, package, class
D. package, class, import
E. import, class, package

Answer: B

NEW QUESTION 74
Which two safely validate inputs? (Choose two.)

A. Delegate numeric range checking of values to the database.
B. Accept only valid characters and input values.
C. Use trusted domain-specific libraries to validate inputs.
D. Assume inputs have already been validated.
E. Modify the input values, as needed, to pass validation.

Answer: AB

NEW QUESTION 77
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);
   }
}
```

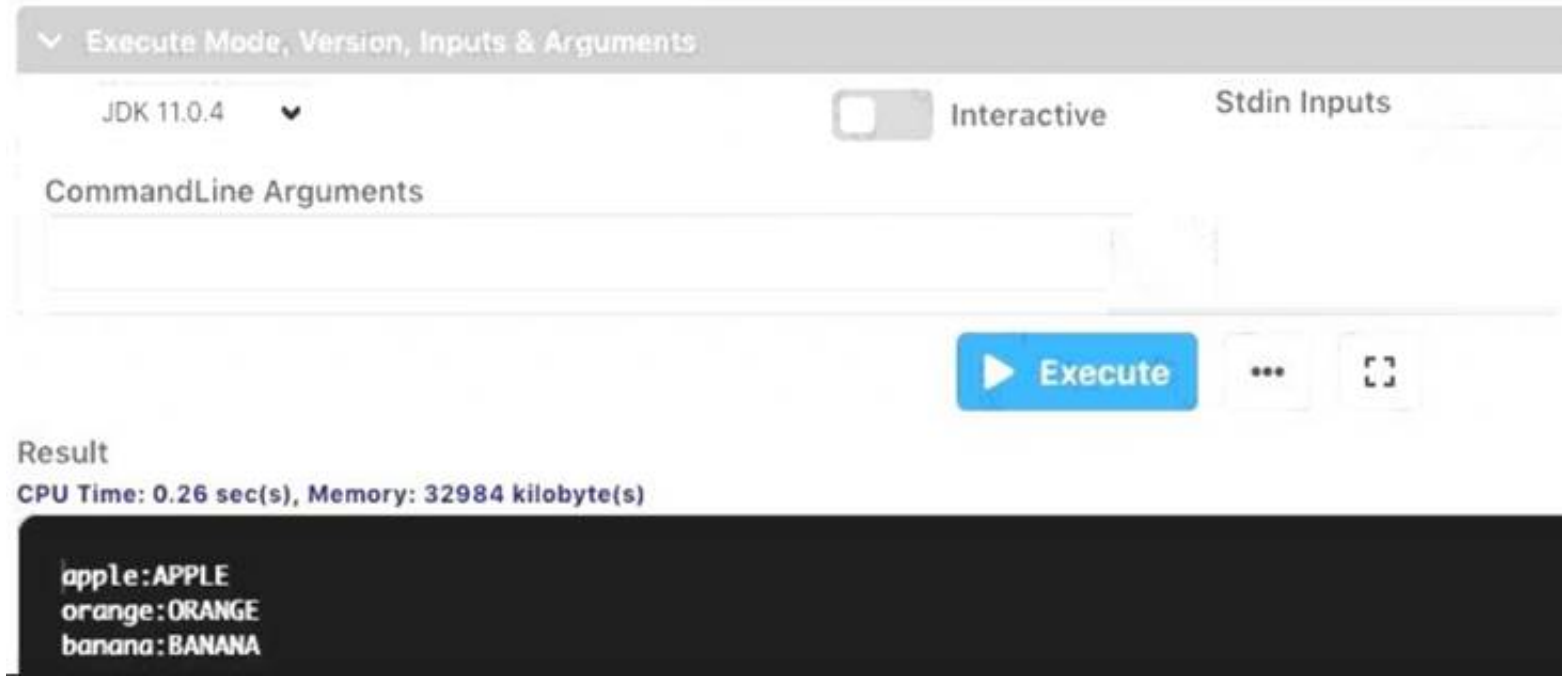
What is the output?

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 }
```



NEW QUESTION 82
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 83
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:

NEW QUESTION 84
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 86
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 87
Which two statements correctly describe capabilities of interfaces and abstract classes? (Choose two.)

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

Answer: AC

NEW QUESTION 88
Given:

```
void myLambda() {
    int i = 25;
    Supplier<Integer> foo = () -> i;
    i++;
    System.out.println(foo.get());
}
```

Which is true?

A. The code compiles but does not print any result.
B. The code prints 25.
C. The code does not compile.
D. The code throws an exception at runtime.

Answer: C

NEW QUESTION 91
Which two statements are correct about modules in Java? (Choose two.)

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

Answer: AC

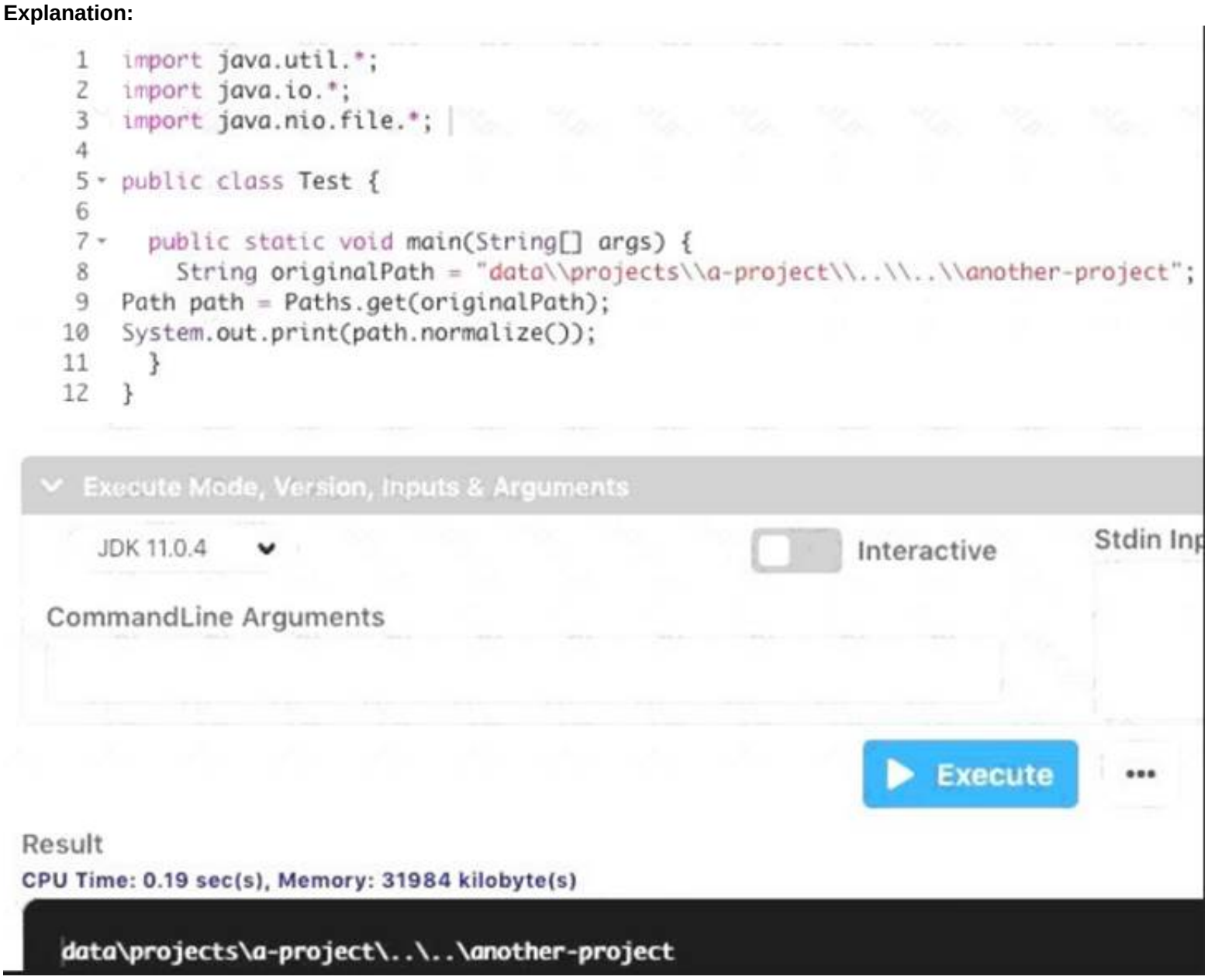
NEW QUESTION 92
Given:
String originalPath = “data\\projects\\a-project\\..\\..\\another-project”; Path path = Paths.get(originalPath); System.out.print(path.normalize());
What is the result?

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

Answer: D

Explanation:

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



NEW QUESTION 93

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

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

What does executing this code fragment do?

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

Answer: C

NEW QUESTION 95
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 98
Given:

```
try {
   // line 1
   lines.map(l -> l.toUpperCase())
     .forEach (line --> {
        try {
           Files.write(Paths.get("outputFile_to_path"),
line.getBytes(),StandardOpenOption.CREATE);
        } catch (IOExeption 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 99
Given:

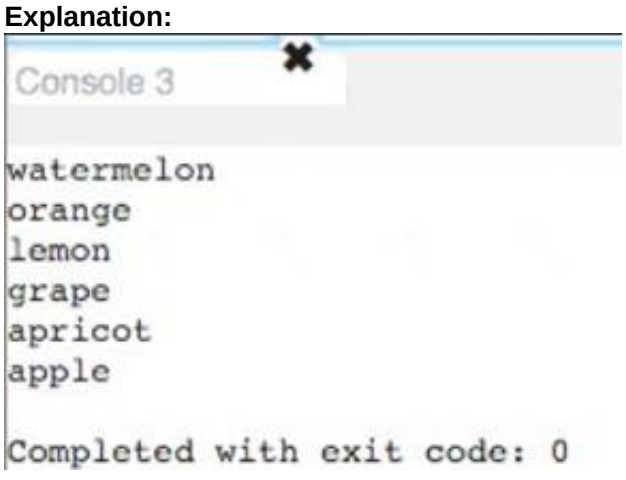
```
import java.util.ArrayList;
import java.util.Arrays;
public class NewMain {
  public static void main(String[] args) {
    String[] fruitNames = { "apple", "orange",
         "grape", "lemon", "apricot", "watermelon" };
    var fruits = new ArrayList<>(Arrays.asList(fruitNames));
    fruits.sort((var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



NEW QUESTION 100
Which two are successful examples of autoboxing? (Choose two.)

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

Answer: AB

NEW QUESTION 105
Given:

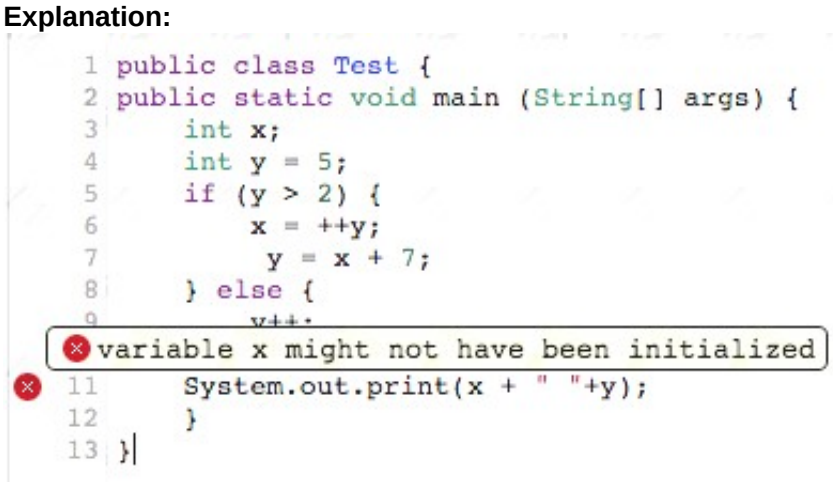
```
public class Test {
    public static void main(String[] args) {
        int x;
        int y = 5;
        if (y > 2) {
            x = ++y;
            y = x + 7;
        } else {
            y++;
        }
        System.out.print(x + " " + y);
    }
}
```

What is the result?

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

Answer: A

Explanation:



NEW QUESTION 106
Which describes an aspect of Java that contributes to high performance?

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

Answer: C

NEW QUESTION 109
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 111
Given:

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

Why does D cause a compilation error?

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

Answer: B

NEW QUESTION 114
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 115
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 119
Given:

```
public method foo() throws FooException {
    ...
}
```

and omitting the throws FooException clause results in a compilation error. Which statement is true about FooException?

A. FooException is a subclass of RuntimeError.
B. FooException is unchecked.
C. The body of foo can only throw FooException.
D. The body of foo can throw FooException or one of its subclasses.

Answer: D

NEW QUESTION 123
Given:

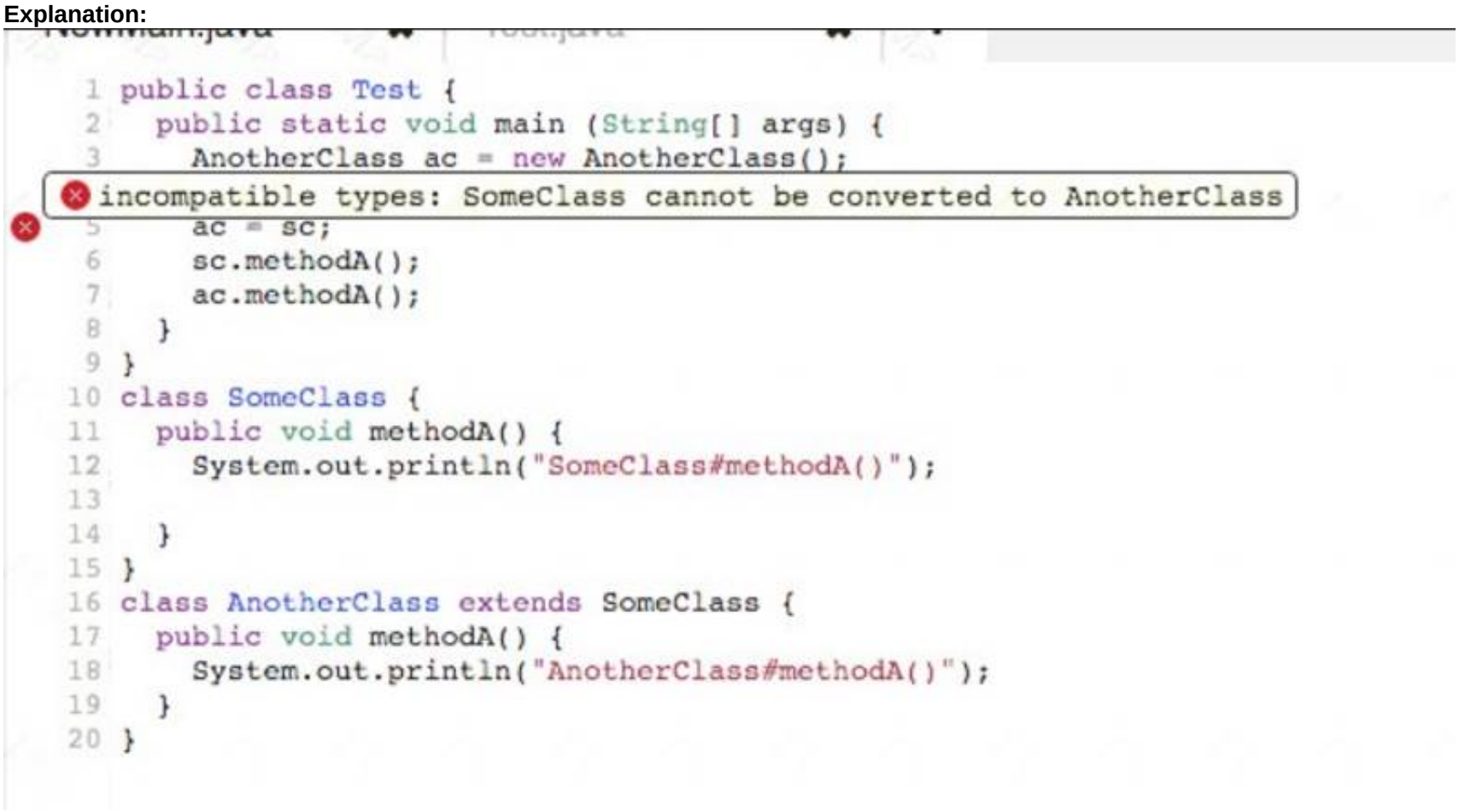
```
public class Test {
  public static void main(String[] args) {
    AnotherClass ac = new AnotherClass();
    SomeClass sc = new AnotherClass();
    ac = sc;
    sc.methodA();
    ac.methodA();
  }
}
class SomeClass {
  public void methodA() {
    System.out.println("SomeClass#methodA()");
  }
}
class 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 124
Given:
var data = new ArrayList<>(); data.add(“Peter”);
data.add(30); data.add(“Market Road”); data.set(1, 25); data.remove(2); data.set(3, 1000L); System.out.print(data); What is the output?

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

Answer: D

Explanation:

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

Completed with exit code: 1
```

NEW QUESTION 127
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 129
Given:

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

What is the output?

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

Answer: C

Explanation:

```
14
15    public class Main {
16      public static void main(String[] args) {
17        Optional<String> value = createValue();
18        String str = value.orElse ("Duke");
19        System.out.println(str);
20      }
21      static Optional<String> createValue() {
22        String s = null;
23        return Optional.ofNullable(s);
24      }
25    }
26
```

Result

CPU Time: 0.15 sec(s), Memory: 32572 kilobyte(s)

```
Duke
```

NEW QUESTION 132
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 133
Given:

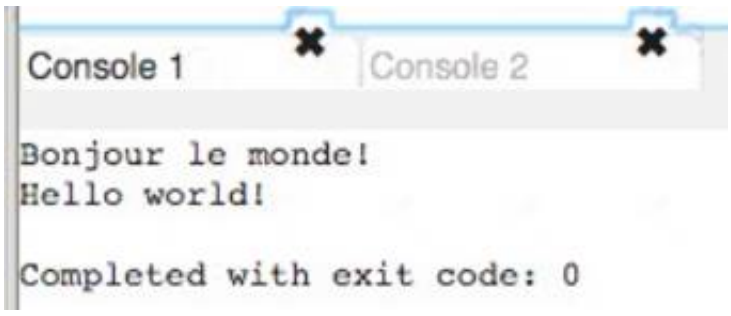
```
public class Foo {
    private void print() {
        System.out.println("Bonjour le monde!");
    }
    public void foo() {
        print();
    }
}

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:

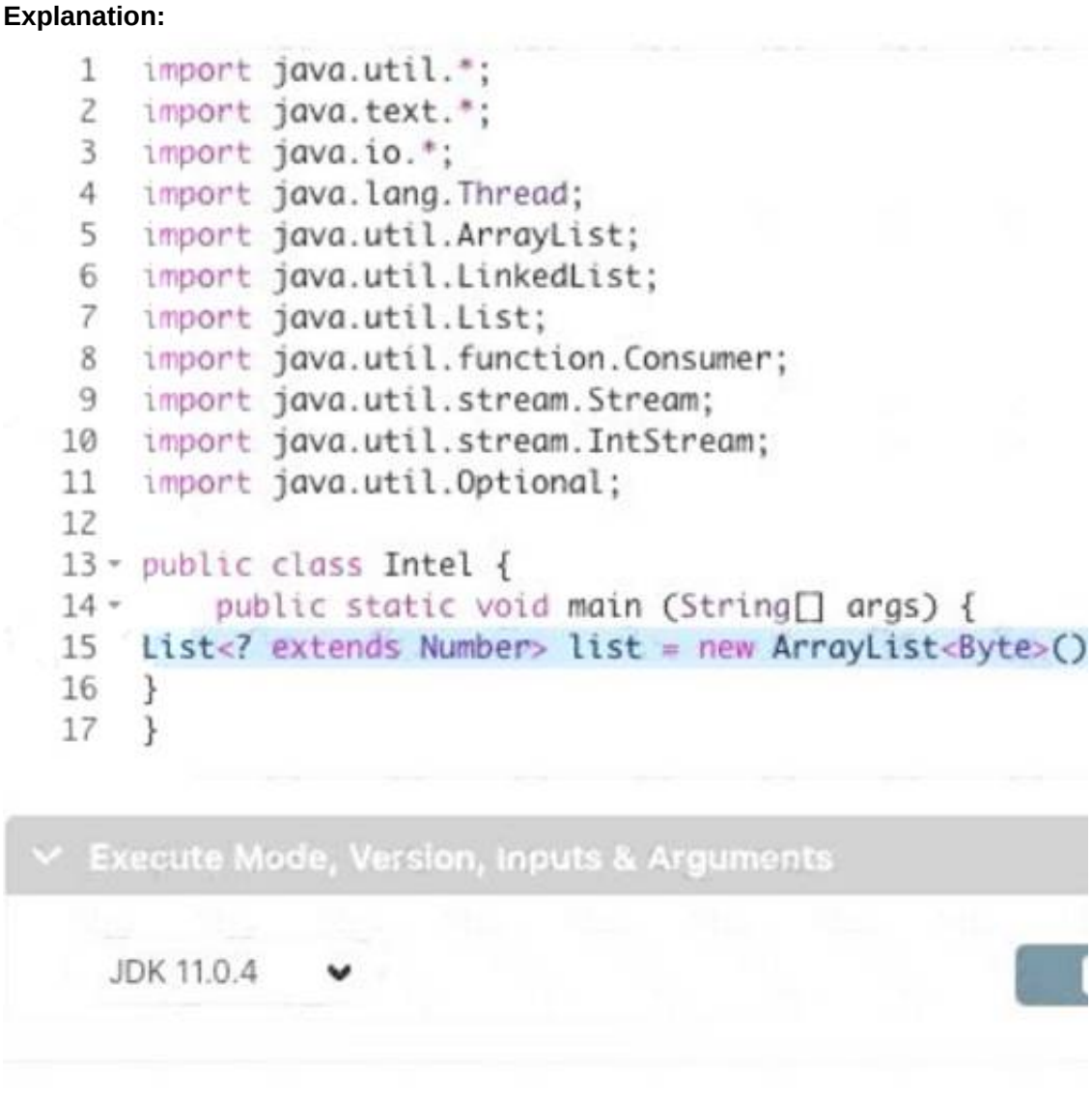


NEW QUESTION 136
Which two statements independently compile? (Choose two.)

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

Answer: AC

Explanation:



NEW QUESTION 140
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 143
Analyze the code:

```
public class Test {
    static String prefix = "Global:";
    private String name = "namescope";
    public static String getName(){
      return new Test().name;
    }
    public static void main(String[] args) {
      Test t = new Test();
      System.out.println(/* Insert code here */);
    }
}
```

Which two options can you insert inside println method to produce Global:namescope? (Choose two.)

A. Test.prefix+Test.name
B. new Test().prefix+new Test().name
C. Test.prefix+Test.getName()
D. Test.getName+prefix
E. prefix+Test.name
F. prefix+name

Answer: BC

NEW QUESTION 145
Given:
/code/a/Test.java containing:

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

and
/code/b/Best.java containing: package b;
public class Best { }
Which is the valid way to generate bytecode for all classes?

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

Answer: E

NEW QUESTION 147
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:

```
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 148
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 151
Which interface in the java.util.function package can return a primitive type?

A. ToDoubleFunction
B. Supplier
C. BiFunction
D. LongConsumer

Answer: A

NEW QUESTION 155
Given:

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

What is the result?

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

Answer: E

NEW QUESTION 156
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 160
Given:

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

and

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

and

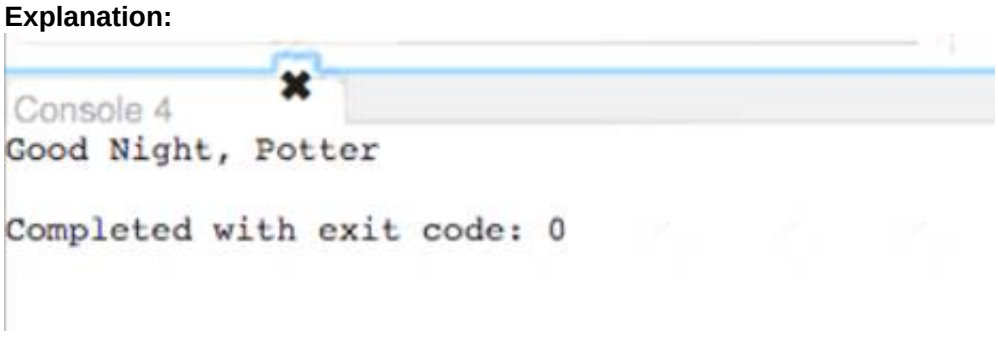
```
class Test {
    public static void main(String[] args) {
        Super s = new Sub();
        System.out.println(s.greeting() + ", " + s.name());
    }
}
```

What is the result?

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

Answer: B

Explanation:



NEW QUESTION 165
Given:

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

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

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

Answer: D

Explanation:

NEW QUESTION 168
Given:

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

What is known about the Sportscar class?

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

Answer: A

NEW QUESTION 173
Given:

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

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

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

Answer: B

NEW QUESTION 177
Given the contents:
MessageBundle.properties file: message=Hello MessageBundle_en.properties file: message=Hello (en) MessageBundle_US.properties file: message=Hello (US)
MessageBundle_en_US.properties file: message=Hello (en_US) MessageBundle_fr_FR.properties file: message=Bonjour
and the code fragment: Locale.setDefault(Locale.FRANCE);
Locale currentLocale = new Locale.Builder().setLanguage("en").build();
ResourceBundle messages = ResourceBundle.getBundle("MessageBundle", currentLocale); System.out. println(messages.getString("message"));
Which file will display the content on executing the code fragment?

A. MessageBundle_en_US.properties
B. MessageBundle_en.properties
C. MessageBundle_fr_FR.properties
D. MessageBundle_US.properties
E. MessageBundle.properties

Answer: C

NEW QUESTION 181
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 184
Given:

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

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

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

Answer: D

NEW QUESTION 189
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 194
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 199
Given:

```
1.   void insertionSort(int values[]) {
2.        int n = values.length;
3.        for (int j = 1; j < n; j++) {
4.             int tmp = values[j];
5.             int i = j - 1;
6.             while ( (i > -1) && (values[i] > tmp) ) {
7.                values[i + 1] = values[i];
8.                  i--;
9.             }
10.            values[i + 1] = tmp;
11.       }
12.  }
```

After which line can we insert assert i < 0 || values[i] <= values[i + 1]; to verify that the values array is partially sorted?

A. after line 8
B. after line 6
C. after line 5
D. after line 10

Answer: B

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
15     void insertionSort (int values[]) {
16       int n = values.length;
17         for (int j = 1; j < n; j++) {
18           int tmp = values[j];
19
20           int i = j - 1;
21           assert i < 0 || values[i] <= values[i + 1];
22           while ((i > 1) && (values[i] > tmp) ) {
23             values[i + 1] = values[i];
24             i--;
25
26           }
27           values[i + 1] = tmp;
28
29         }
30     }
31   }
```

NEW QUESTION 204
Given:

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

What is the result?

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

Answer: A

Explanation:

```
1
2  public class Price {
3    private final double value;
4    public Price(String value) {
5      this(Double.parseDouble (value));
6    }
7    public Price(double value) {
8      this.value = value;
   variable value might not have been initialized
10   public Price (){}
11   public double getValue() { return value; }
12   public static void main (String[] args) {
13     Price p1 = new Price("1.99");
14     Price p2 = new Price("2.99");
15     Price p3 = new Price();
16     System.out.println(p1.getValue()+","+p2.getValue()+","+p3.getValue());
17   }
18 }
```

NEW QUESTION 208
Given:

```
import java.util.*;

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

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

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

What is the result?

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

Answer: D

Explanation:

```
Console 1
Map: 4 Keys: 0Values: 0

Completed with exit code: 0
```

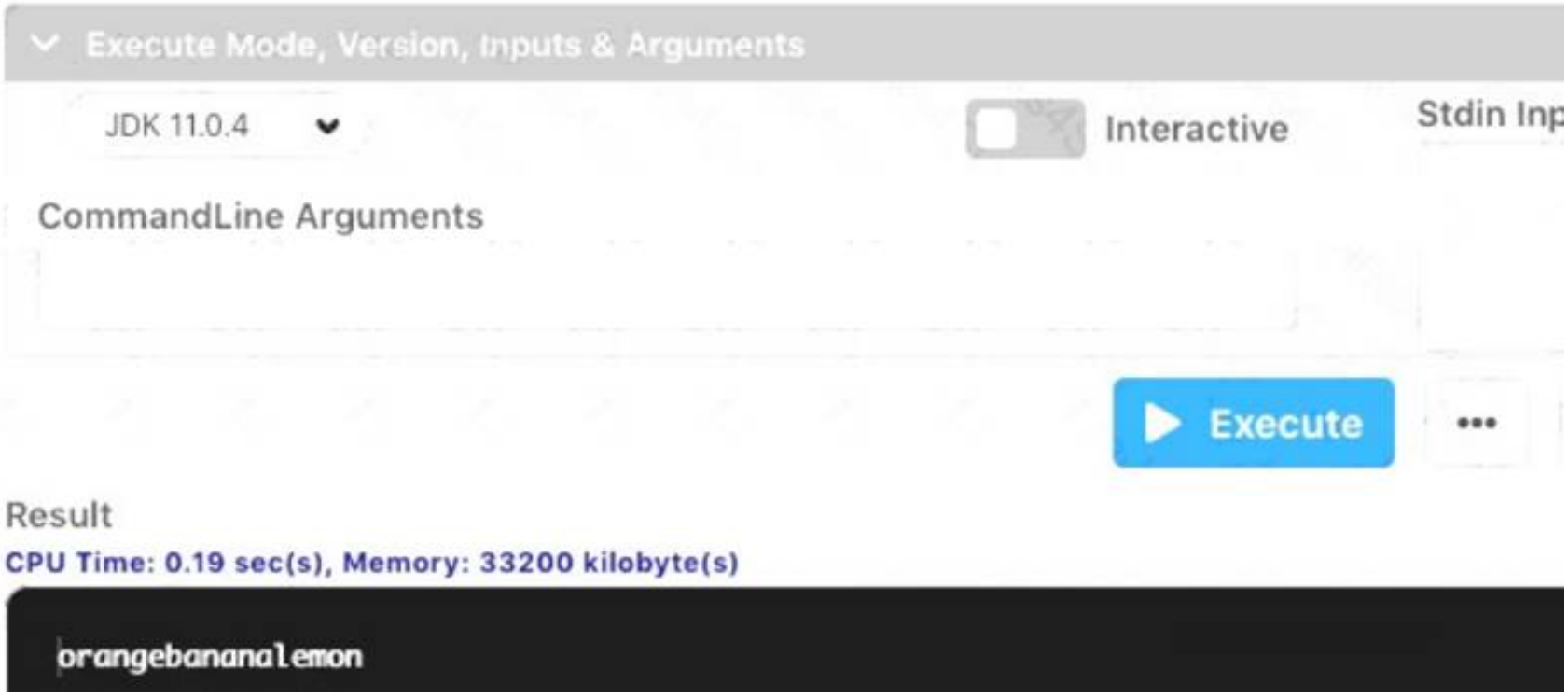
NEW QUESTION 212
Given:
var fruits = List.of(“apple”, “orange”, “banana”, “lemon”);
You want to examine the first element that contains the character n. Which statement will accomplish this?

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
```



NEW QUESTION 213
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.printin(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 214
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 217
.....

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