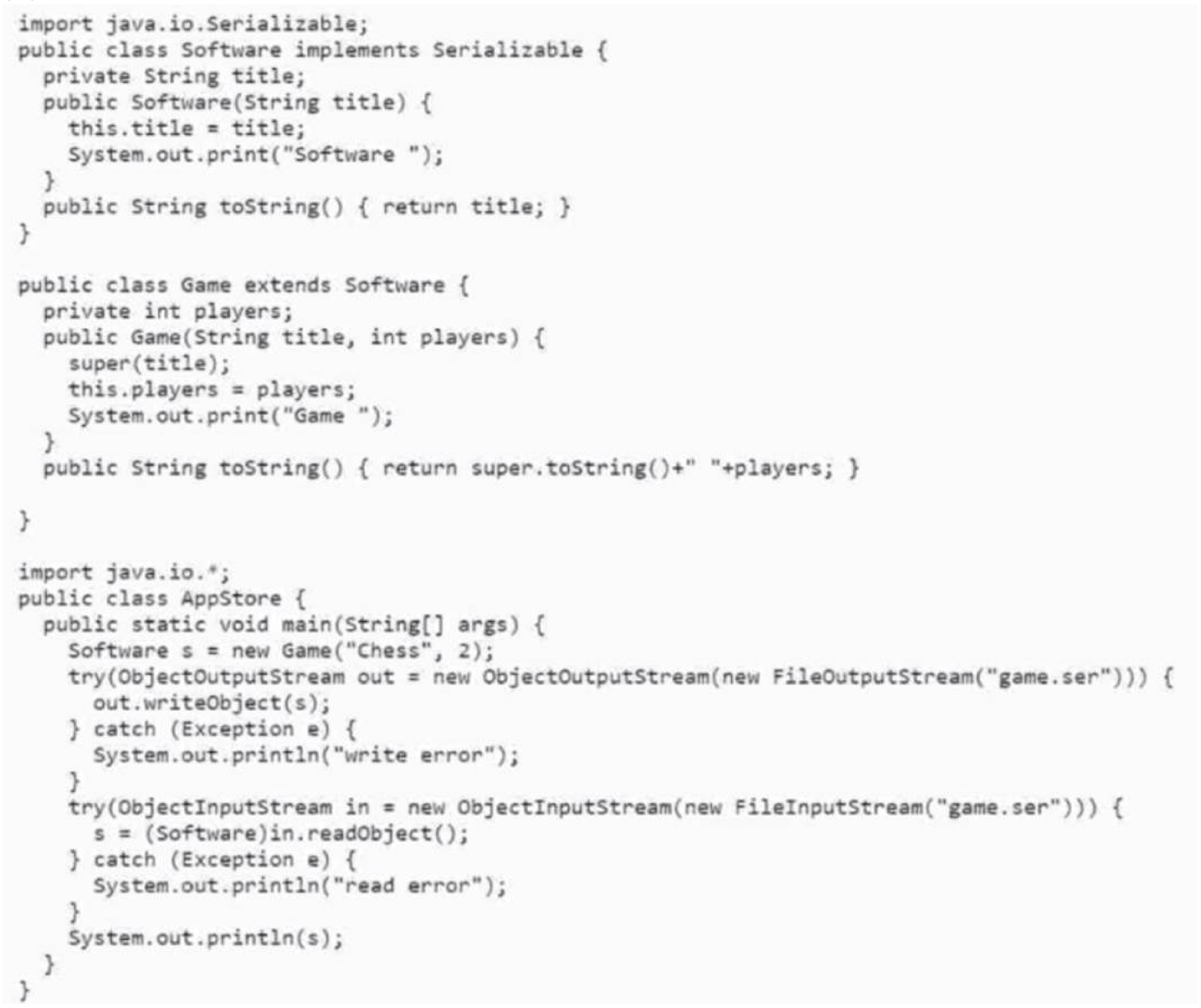
Exam Questions 1z0-829

Java SE 17 Developer

https://www.2passeasy.com/dumps/1z0-829/

NEW QUESTION 1
Given:



```
import java.io.Serializable;
public class Software implements Serializable {
  private String title;
  public Software(String title) {
    this.title = title;
    System.out.print("Software ");
  }
  public String toString() { return title; }
}

public class Game extends Software {
  private int players;
  public Game(String title, int players) {
    super(title);
    this.players = players;
    System.out.print("Game ");
  }
  public String toString() { return super.toString()+" "+players; }

}

import java.io.*;
public class AppStore {
  public static void main(String[] args) {
    Software s = new Game("Chess", 2);
    try(ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream("game.ser"))) {
      out.writeObject(s);
    } catch (Exception e) {
      System.out.println("write error");
    }
    try(ObjectInputStream in = new ObjectInputStream(new FileInputStream("game.ser"))) {
      s = (Software)in.readObject();
    } catch (Exception e) {
      System.out.println("read error");
    }
    System.out.println(s);
  }
}
```

What is the result?

A. Software Game Chess 0
B. Software Game Software Game Chese 2
C. Software game write error
D. Software Game Software Game chess 0
E. Software Game Chess 2
F. Software Game read error

Answer: B

Explanation:
The answer is B because the code uses the writeObject and readObject methods of
the ObjectOutputStream and ObjectInputStream classes to serialize and deserialize the Game object. These methods use the default serialization mechanism, which writes and reads the state of the object??s fields, including the inherited ones. Therefore, the title field of the Software class is also serialized and deserialized along with the players field of the Game class. The toString method of the Game class calls the toString method of the Software class using super.toString(), which returns the value of the title field. Hence, when the deserialized object is printed, it shows ??Software Game Software Game Chess 2??.
References:
? Oracle Certified Professional: Java SE 17 Developer
? Java SE 17 Developer
? OCP Oracle Certified Professional Java SE 17 Developer Study Guide
? Serialization and Deserialization in Java with Example

NEW QUESTION 2
Which statement is true?

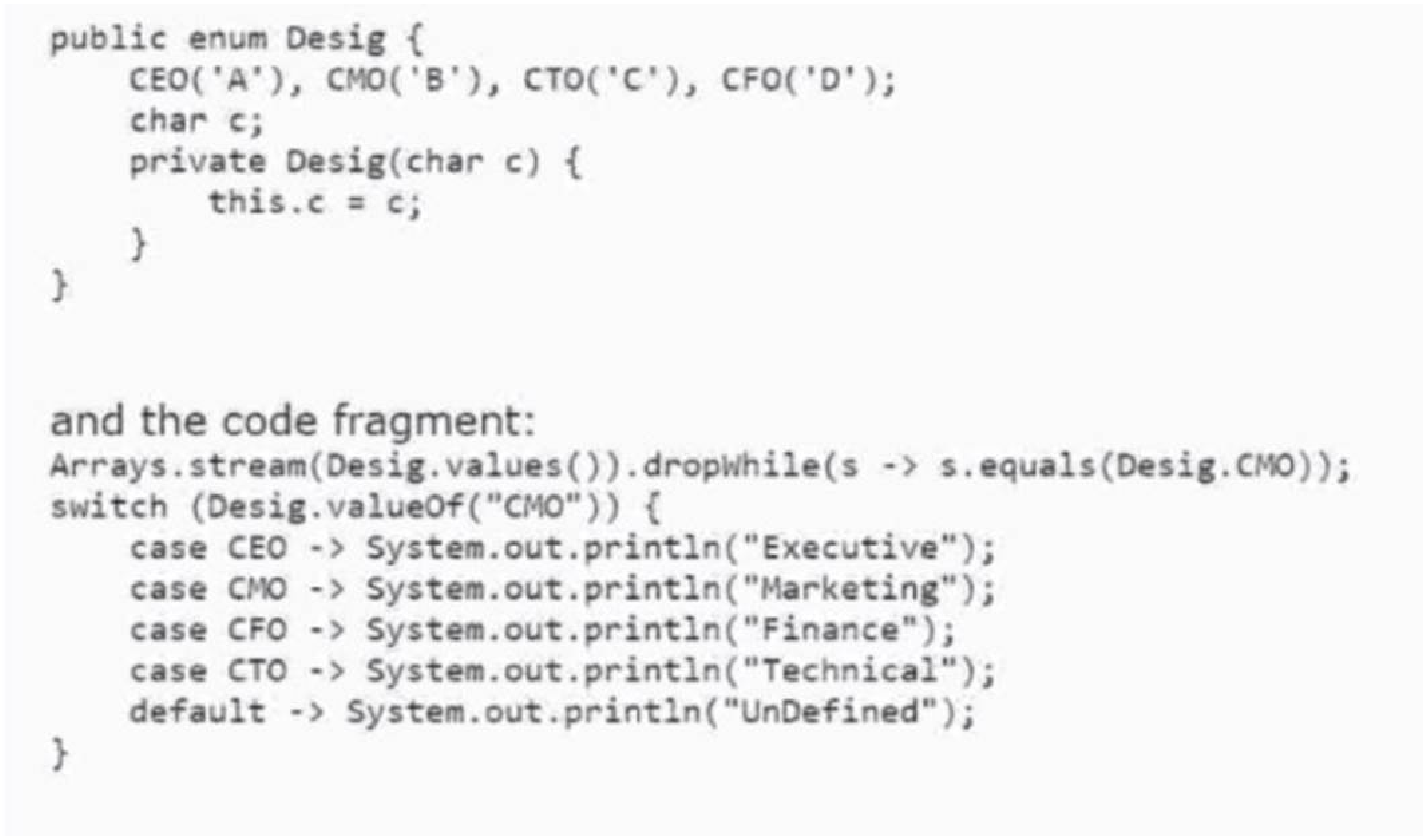
A. The tryLock () method returns a boolean indicator immediately regardless if it has or has not managed to acquire the lock.
B. The tryLock () method returns a boolean indicator immediately if it has managed to acquire the lock, otherwise it waits for the lock acquisition.
C. The lock () method returns a boolean indicator immediately if it has managed to acquire the lock, otherwise it waits for the lock acquisition.

D. The Lock () method returns a boolean indicator immediately regardless if it has or has not managed to acquire the lock

Answer: A

Explanation:
The tryLock () method of the Lock interface is a non-blocking attempt to acquire a lock. It returns true if the lock is available and acquired by the current thread, and false otherwise. It does not wait for the lock to be released by another thread. This is different from the lock () method, which blocks the current thread until the lock is acquired, and does not return any value. References: https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/concurrent/locks/Loc k.html#tryLock(), 3, 4, 5

NEW QUESTION 3
Given:



```
public enum Desig {
    CEO('A'), CMO('B'), CTO('C'), CFO('D');
    char c;
    private Desig(char c) {
        this.c = c;
    }
}
```

and the code fragment:

```
Arrays.stream(Desig.values()).dropWhile(s -> s.equals(Desig.CMO));
switch (Desig.valueOf("CMO")) {
    case CEO -> System.out.println("Executive");
    case CMO -> System.out.println("Marketing");
    case CFO -> System.out.println("Finance");
    case CTO -> System.out.println("Technical");
    default -> System.out.println("UnDefined");
}
```

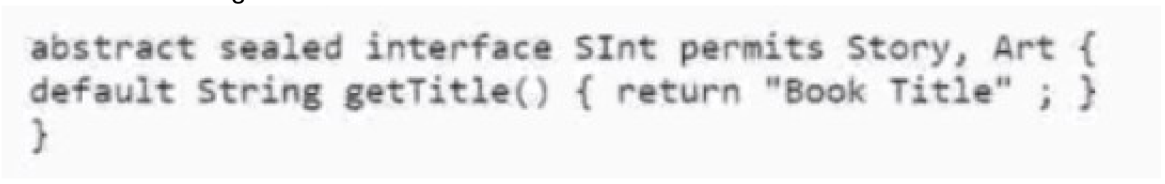
What is the result

A. Marketing Finance Technical
B. Marketing Undefined
C. UnDefined
D. Marketing

Answer: C

Explanation:
The code fragment is using the switch statement with the new Java 17 syntax. The switch statement checks the value of the variable desig and executes the corresponding case statement. In this case, the value of desig is ??CTO??, which does not match any of the case labels. Therefore, the default case statement is executed, which prints ??Undefined??. The other case statements are not executed, because there is no fall through in the new syntax. Therefore, the output of the code fragment is: Undefined

NEW QUESTION 4
Given the code fragment:



```
abstract sealed interface SInt permits Story, Art {
default String getTitle() { return "Book Title" ; }
}
```

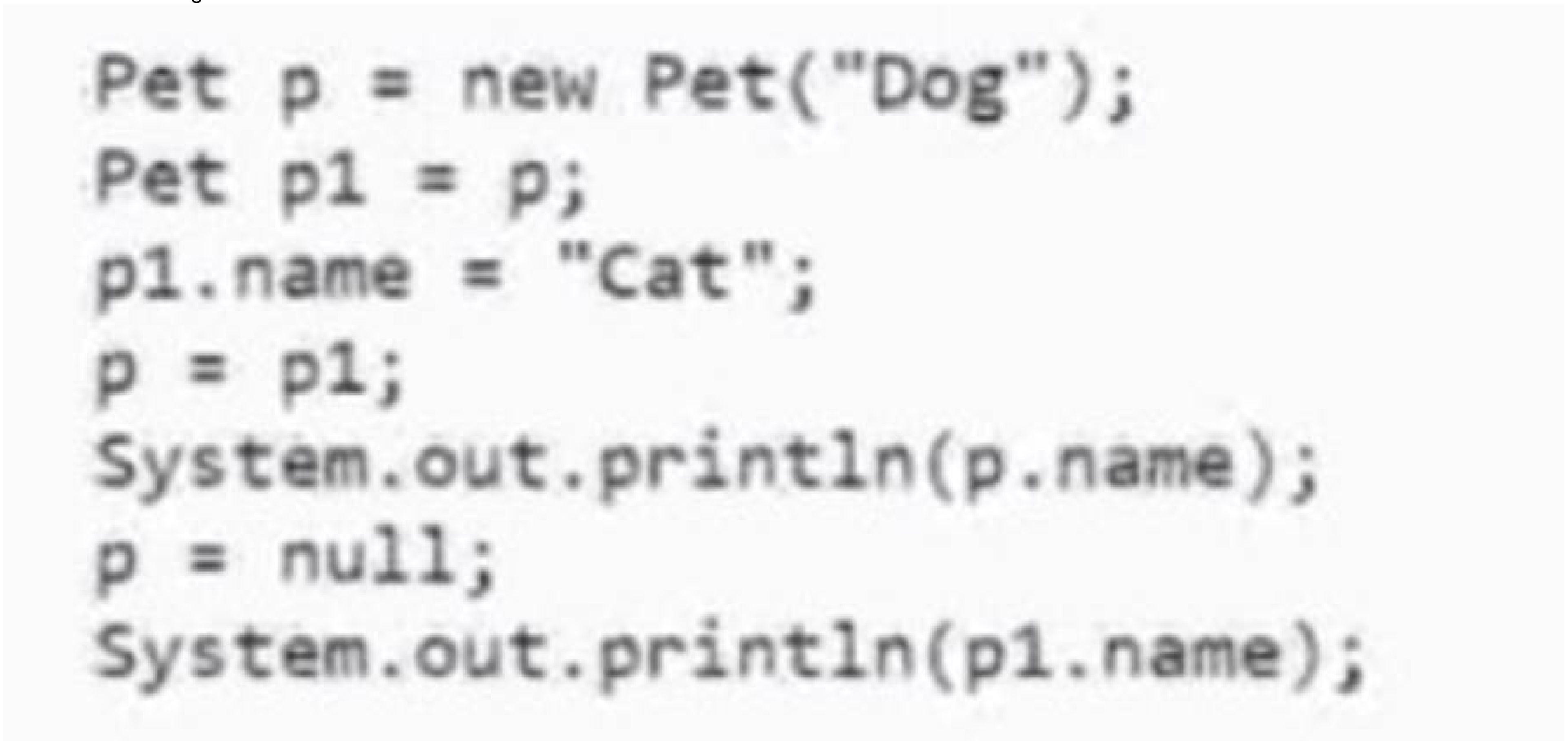
abstract sealed interface SInt permits Story, Art { default String getTitle() { return "Book Title" ; }
}
Which set of class definitions compiles?

A. Interace story extends STnt {} Interface Art extends SInt {}
B. Public interface story extends sInd {} Public interface Art extends SInt {}
C. Sealed interface Storty extends sInt {} Non-sealed class Art implements Sint {}
D. Non-sealed interface story extends SInt {} Class Art implements Sint {}
E. Non-sealed interface story extends SInt {} Non-sealed interaface Art extends Sint {}

Answer: C

Explanation:
The answer is C because the code fragment given is an abstract sealed interface SInt that permits Story and Art. The correct answer is option C, which is a sealed interface Story that extends SInt and a non-sealed class Art that implements SInt. This is because a sealed interface can only be extended by the classes or interfaces that it permits, and a non-sealed class can implement a sealed interface.
Option A is incorrect because interface is misspelled as interace, and Story and Art should be capitalized as they are the names of the permitted classes or interfaces.
Option B is incorrect because public is misspelled as public, and sInd should be SInt as it is the name of the sealed interface.
Option D is incorrect because a non-sealed interface cannot extend a sealed interface, as it would violate the restriction of permitted subtypes.
Option E is incorrect because both Story and Art cannot be non-sealed interfaces, as they
would also violate the restriction of permitted subtypes.
References:
? Oracle Certified Professional: Java SE 17 Developer
? Java SE 17 Developer
? OCP Oracle Certified Professional Java SE 17 Developer Study Guide
? Sealed Classes and Interfaces in Java 15 | Baeldung
? Sealed Class in Java - Javatpoint

NEW QUESTION 5
Given the code fragment:



```
Pet p = new Pet("Dog");
Pet p1 = p;
p1.name = "Cat";
p = p1;
System.out.println(p.name);
p = null;
System.out.println(p1.name);
```

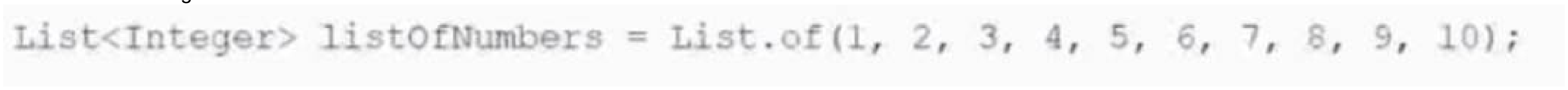
What is the result?

A. A.Cat Dog
B. A NullPointerException is thrown CatCat
C. Dog Dog
D. Cat null

Answer: D

Explanation:
The answer is E because the code fragment creates a new Pet object with the name ??Dog?? and assigns it to the variable p. Then, it assigns p to p1. Next, it changes the name of p1 to ??Cat??. Then, it assigns p1 to p. Finally, it sets p to null and prints the name of p and p1. The output will be ??Cat?? and ??null?? because p is set to null and p1 still points to the Pet object with the name ??Cat??.

NEW QUESTION 6
Given the code fragment:



```
List<Integer> listOfNumbers = List.of(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
```

Which code fragment returns different values?

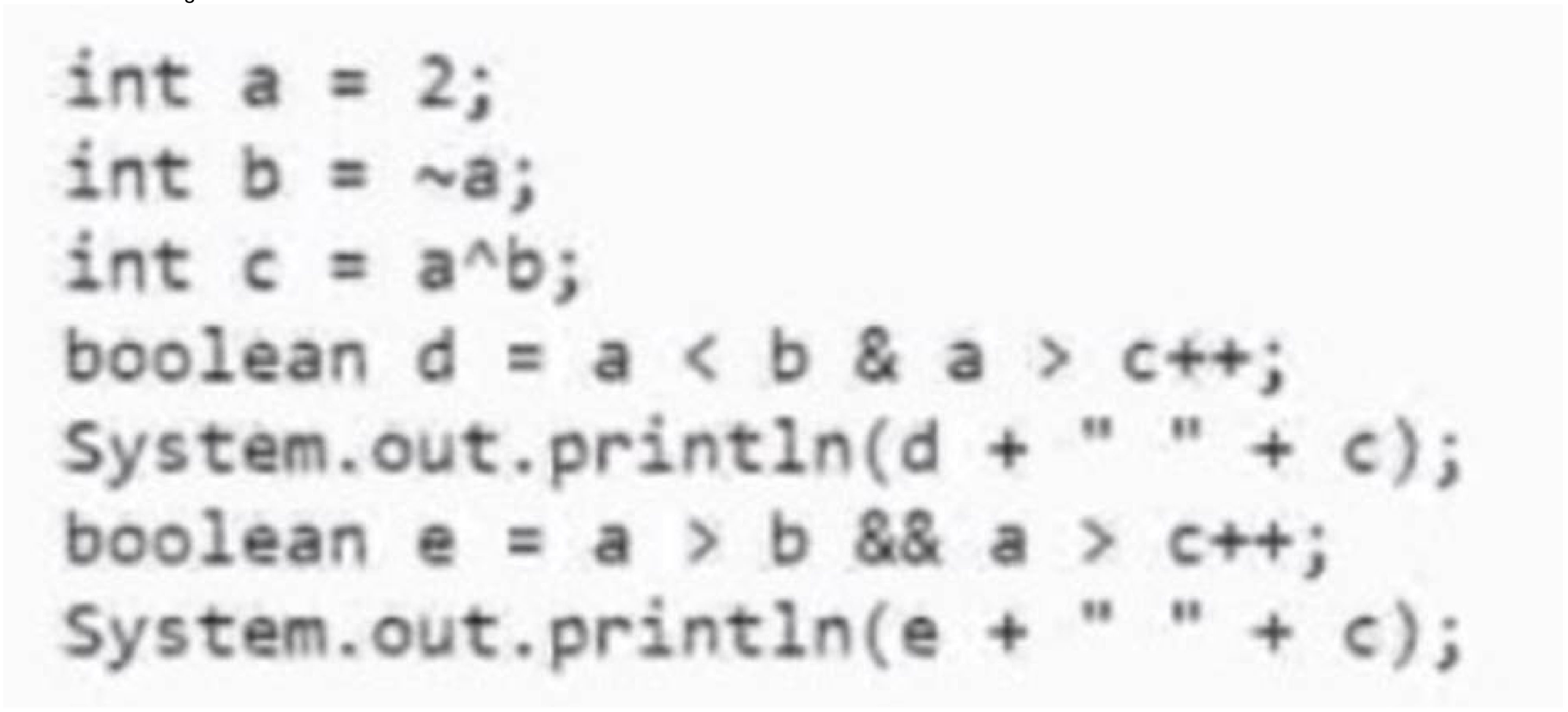
A. int sum = listOfNumber
B. parallelStream () reduce (5, Integer:: sum) ;
C. int sum = listOfNumber
D. Stream () reduce (5, (a, b) -> a + b) ;
E. int sum = listOfNumber
F. Stream () reduce (Integer:: sum) ; +5;
G. int sum = listOfNumber
H. parallelStream () reduce ({m, n) -> m +n) orElse (5) +5;
I. int sum = listOfNumber
J. Stream () reduce (0, Integer:: sum) + 5

Answer: C

Explanation:

The answer is C because the code fragment uses a different syntax and logic for the reduce operation than the other options. The reduce method in option C takes a single parameter, which is a BinaryOperator that combines two elements of the stream into one. The method returns an Optional, which may or may not contain a value depending on whether the stream is empty or not. The code fragment then adds 5 to the result of the reduce method, regardless of whether it is present or not. This may cause an exception if the Optional is empty, or produce a different value than the other options if the Optional is not empty.
The other options use a different syntax and logic for the reduce operation. They all take two parameters, which are an identity value and a BinaryOperator that combines an element of the stream with an accumulator. The method returns the final accumulator value, which is equal to the identity value if the stream is empty, or the result of applying the BinaryOperator to all elements of the stream otherwise. The code fragments then add 5 to the result of the reduce method, which will always produce a valid value.
For example, suppose listOfNumbers contains [1, 2, 3]. Then, option A will perform the following steps:
? Initialize accumulator to identity value 5
? Apply BinaryOperator Integer::sum to accumulator and first element: 5 + 1 = 6
? Update accumulator to 6
? Apply BinaryOperator Integer::sum to accumulator and second element: 6 + 2 = 8
? Update accumulator to 8
? Apply BinaryOperator Integer::sum to accumulator and third element: 8 + 3 = 11
? Update accumulator to 11
? Return final accumulator value 11
? Add 5 to final accumulator value: 11 + 5 = 16
Option B will perform the same steps as option A, except using a lambda expression instead of a method reference for the BinaryOperator. Option D will perform the same steps as option A, except using parallelStream instead of stream, which may change the order of applying the BinaryOperator but not the final result.
Option E will perform the same steps as option A, except using identity value 0 instead of 5.
Option C, however, will perform the following steps:
? Apply BinaryOperator Integer::sum to first and second element: 1 + 2 = 3
? Apply BinaryOperator Integer::sum to previous result and third element: 3 + 3 = 6
? Return Optional containing final result value 6
? Add 5 to Optional value: Optional.of(6) + 5 = Optional.of(11)
As you can see, option C produces a different value than the other options, and also uses a different syntax and logic for the reduce operation. References:
? Oracle Certified Professional: Java SE 17 Developer
? Java SE 17 Developer
? OCP Oracle Certified Professional Java SE 17 Developer Study Guide
? Guide to Stream.reduce()

NEW QUESTION 7
Given the code fragment:



```
int a = 2;
int b = ~a;
int c = a^b;
boolean d = a < b & a > c++;
System.out.println(d + " " + c);
boolean e = a > b && a > c++;
System.out.println(e + " " + c);
```

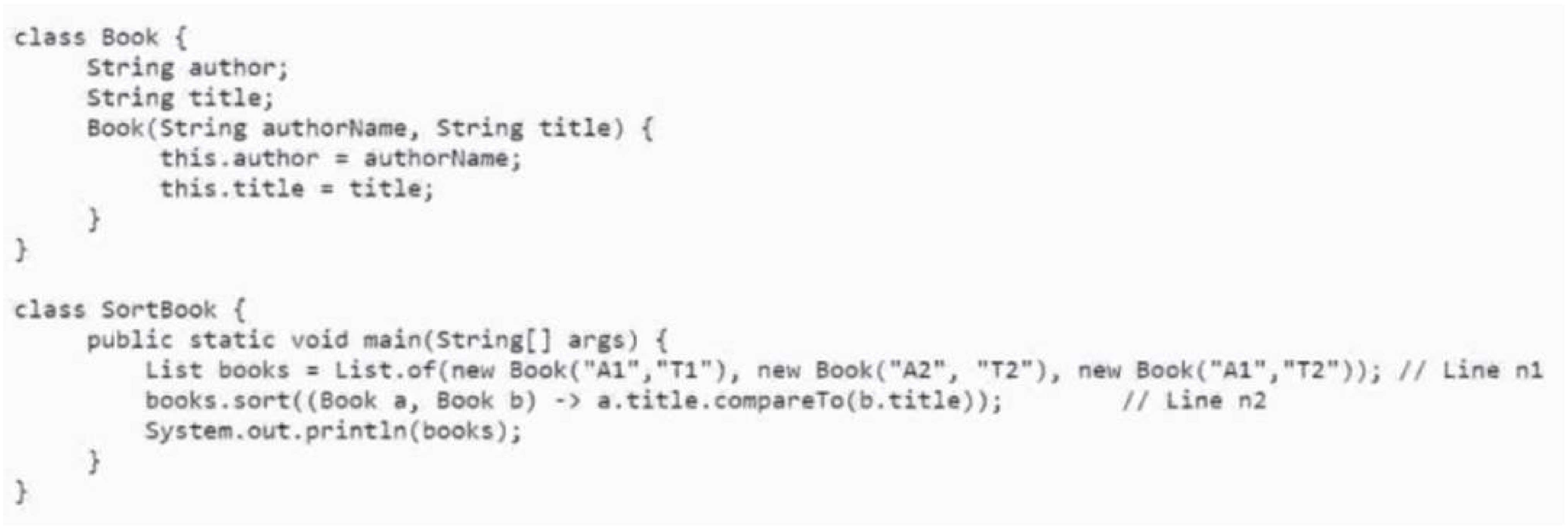
What is the result?

A. false 1false 2
B. true 1false 2
C. false 1ture 2
D. falase 0true 1

Answer: B

Explanation:
The code fragment is comparing the values of a, b, and c using the < and > operators. The first comparison, d, is checking if a is less than b and greater than c. Since a is equal to 2, b is equal to -2, and c is equal to -4, this comparison will evaluate to true. The second comparison, e, is checking if a is greater than b and a is greater than c. Since a is equal to 2, b is equal to -2, and c is equal to -4, this comparison will evaluate to false.
Therefore, the result will be true 1 false 2. References: Operators (The Java™ Tutorials > Learning the Java Language - Oracle

NEW QUESTION 8
Given the code fragment:



```
class Book {
     String author;
     String title;
     Book(String authorName, String title) {
          this.author = authorName;
          this.title = title;
     }
}

class SortBook {
     public static void main(String[] args) {
         List books = List.of(new Book("A1","T1"), new Book("A2", "T2"), new Book("A1","T2")); // Line n1
         books.sort((Book a, Book b) -> a.title.compareTo(b.title));        // Line n2
         System.out.println(books);
     }
}
```

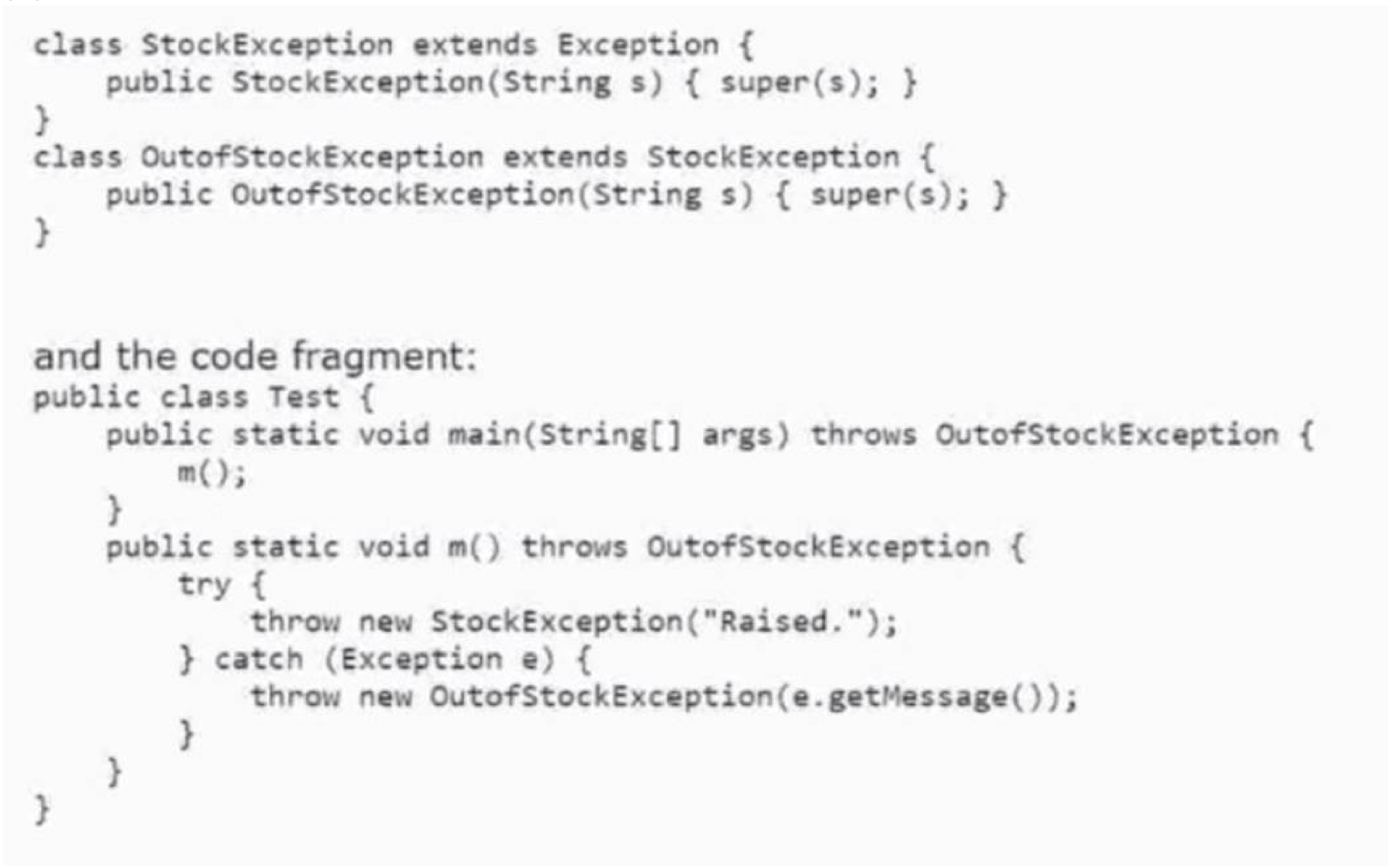
Which action sorts the book list?

A. At Line n2, replace books,sort() with books.stream().sort(0.
B. At line n1, convert books type to mutable ArrayList type.
C. At Line n1, convert type to mutable array type.
D. At Line n2, replace compareTo () with compare ().

Answer: D

Explanation:
The code fragment is trying to sort a list of books using the Collections.sort() method. The correct answer is D, because the compareTo() method is not the correct way to compare two objects in a Comparator. The compare() method is the correct way to compare two objects in a Comparator and return an int value that indicates their order1. The compareTo() method is used to implement the Comparable interface, which defines the natural order of objects of a class2. The other options are incorrect because they either do not change the type of the list, which is already mutable, or they do not use the correct syntax for sorting a stream, which requires a terminal operation such as collect()3. References: Comparator (Java SE 17 & JDK 17), Comparable (Java SE 17 & JDK 17), Stream (Java SE 17 & JDK 17)

NEW QUESTION 9
Given:



```
class StockException extends Exception {
     public StockException(String s) { super(s); }
}
class OutofStockException extends StockException {
     public OutofStockException(String s) { super(s); }
}
```

and the code fragment:

```
public class Test {
     public static void main(String[] args) throws OutofStockException {
          m();
     }
     public static void m() throws OutofStockException {
          try {
               throw new StockException("Raised.");
          } catch (Exception e) {
               throw new OutofStockException(e.getMessage());
          }
     }
}
```

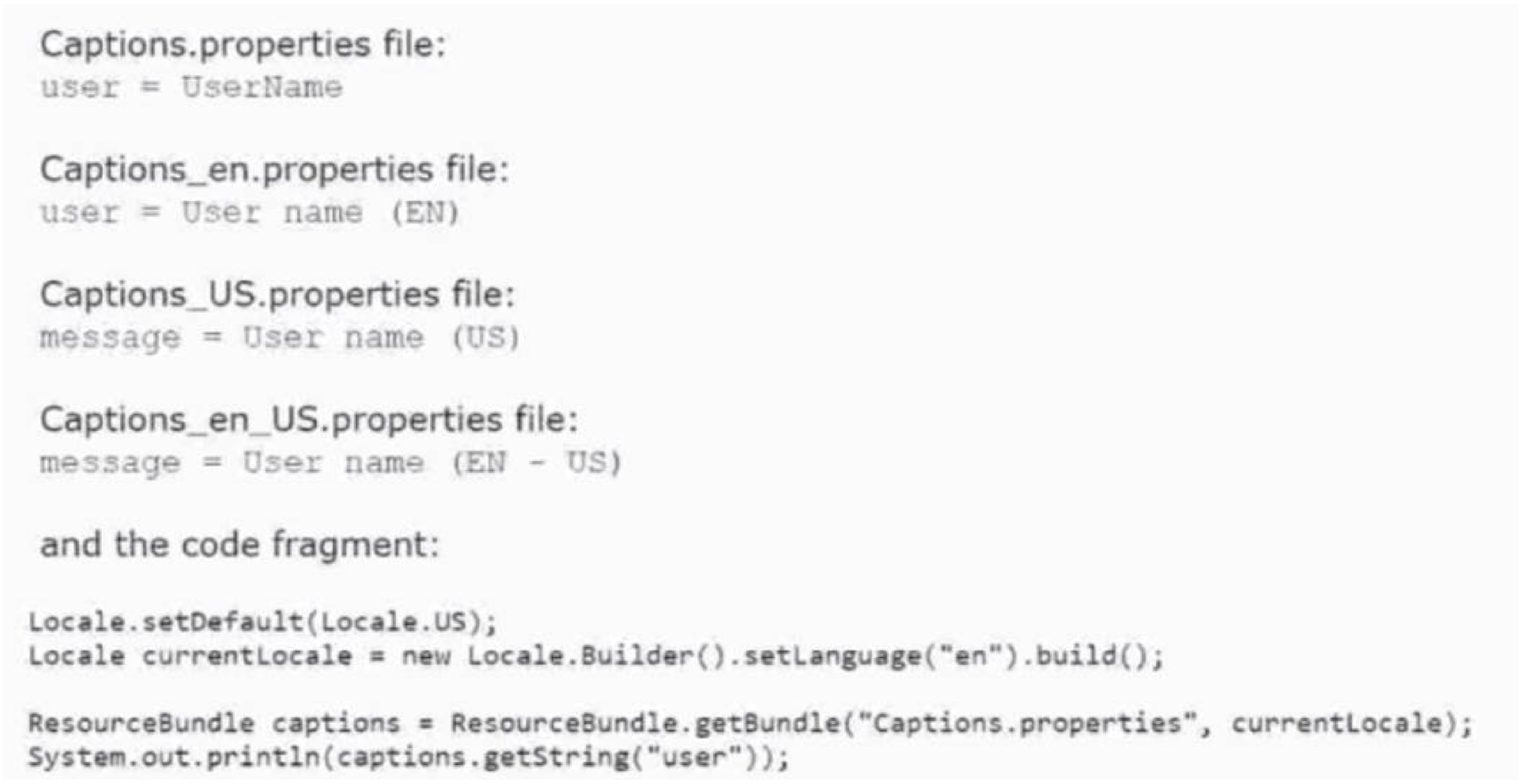
Which statement is true?

A. The program throws StockException.
B. The program fails to compile.
C. The program throws outofStockException.
D. The program throws ClassCastException

Answer: B

Explanation:
The answer is B because the code fragment contains a syntax error that prevents it from compiling. The code fragment tries to catch a StockException in line 10, but the catch block does not have a parameter of type StockException. The catch block should have a parameter of type StockException, such as:
catch (StockException e) { // handle the exception }
This is required by the Java syntax for the catch clause, which must have a parameter that is a subclass of Throwable. Without a parameter, the catch block is invalid and causes a compilation error.
Option A is incorrect because the program does not throw a StockException, as it does not compile.
Option C is incorrect because the program does not throw an OutofStockException, as it does not compile.
Option D is incorrect because the program does not throw a ClassCastException, as it does not compile. References:
? Oracle Certified Professional: Java SE 17 Developer
? Java SE 17 Developer
? OCP Oracle Certified Professional Java SE 17 Developer Study Guide
? The try-with-resources Statement (The Java™ Tutorials > Essential Classes > Exceptions)
? The catch Blocks (The Java™ Tutorials > Essential Classes > Exceptions)

NEW QUESTION 10
Given:



Captions.properties file:
```
user = UserName
```

Captions_en.properties file:
```
user = User name (EN)
```

Captions_US.properties file:
```
message = User name (US)
```

Captions_en_US.properties file:
```
message = User name (EN - US)
```

and the code fragment:

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

ResourceBundle captions = ResourceBundle.getBundle("Captions.properties", currentLocale);
System.out.println(captions.getString("user"));
```

What is the result?

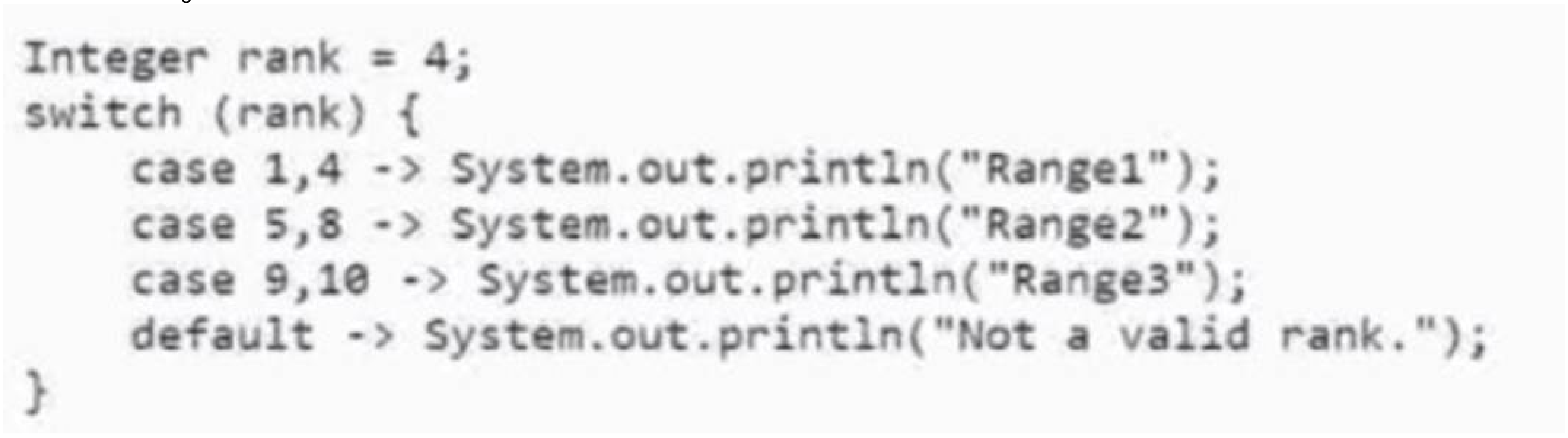
A. User name (US)
B. The program throws a MissingResourceException.
C. User name (EN – US)
D. UserName
E. User name (EN)

Answer: B

Explanation:
The answer is B because the code fragment contains a logical error that causes a MissingResourceException at runtime. The code fragment tries to load a resource bundle with the base name ??Captions.properties?? and the locale ??en_US??. However, there is no such resource bundle available in the classpath. The available resource bundles are:
? Captions.properties
? Captions_en.properties
? Captions_US.properties
? Captions_en_US.properties
The ResourceBundle class follows a fallback mechanism to find the best matching resource bundle for a given locale. It first tries to find the resource bundle with the exact locale, then it tries to find the resource bundle with the same language and script, then it tries to find the resource bundle with the same language, and finally it tries to find the default resource bundle with no locale. If none of these resource bundles are found, it throws a MissingResourceException.
In this case, the code fragment is looking for a resource bundle with the base name ??Captions.properties?? and the locale ??en_US??. The ResourceBundle class will try to find the following resource bundles in order:
? Captions.properties_en_US
? Captions.properties_en
? Captions.properties
However, none of these resource bundles exist in the classpath. Therefore, the ResourceBundle class will throw a MissingResourceException.
To fix this error, the code fragment should use the correct base name of the resource bundle family, which is ??Captions?? without the ??.properties?? extension. For example: ResourceBundle captions = ResourceBundle.getBundle(??Captions??, currentLocale); This will load the appropriate resource bundle for the current locale, which is ??Captions_en_US.properties?? in this case. References:
? Oracle Certified Professional: Java SE 17 Developer
? Java SE 17 Developer
? OCP Oracle Certified Professional Java SE 17 Developer Study Guide

? ResourceBundle (Java Platform SE 8)
? About the ResourceBundle Class (The Java™ Tutorials > Internationalization)

NEW QUESTION 10
Given the code fragment:



```
Integer rank = 4;
switch (rank) {
    case 1,4 -> System.out.println("Range1");
    case 5,8 -> System.out.println("Range2");
    case 9,10 -> System.out.println("Range3");
    default -> System.out.println("Not a valid rank.");
}
```

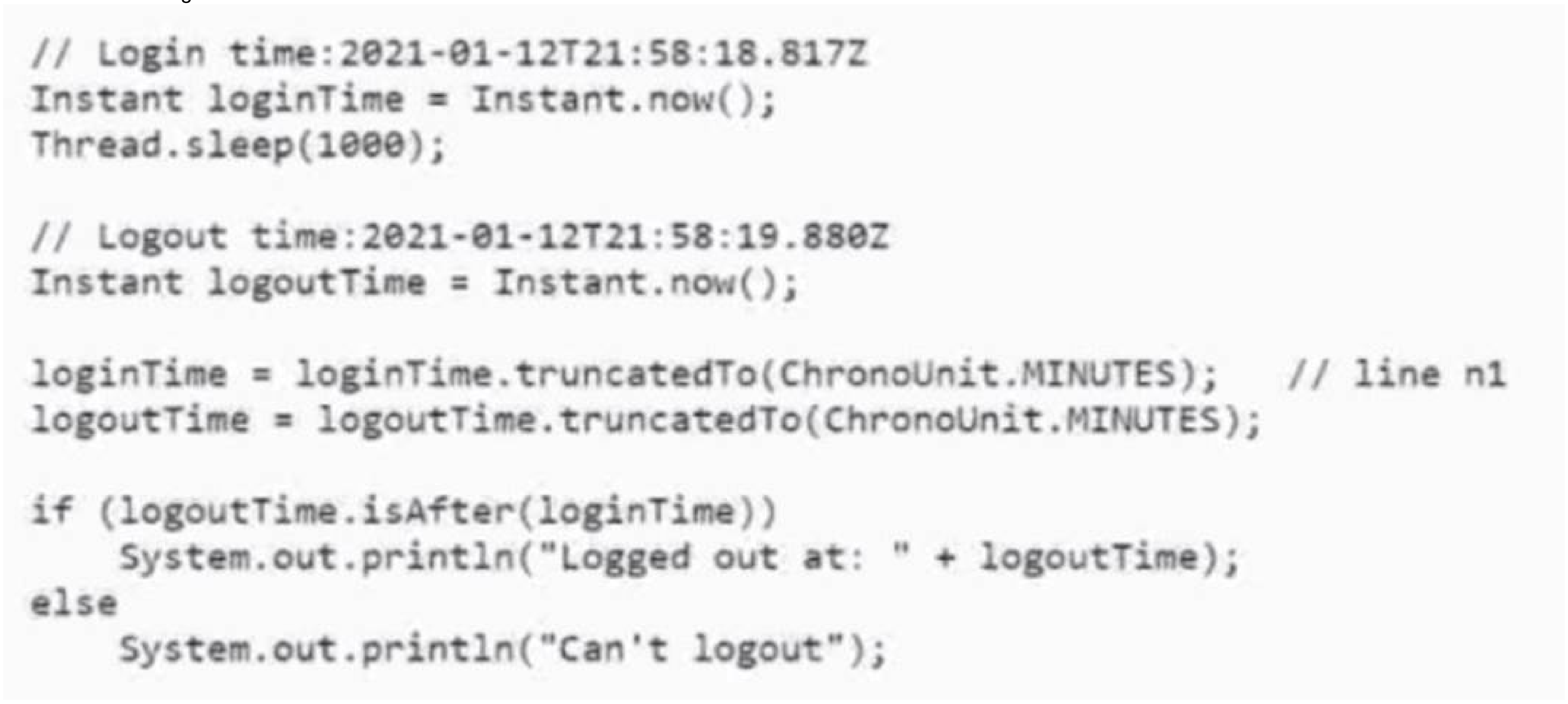
What is the result?

A. Range 1Range 2Range 3
B. Range1Note a valid rank.
C. Range 1Range 2Range 3Range 1Not a valida rank
D. Range 1

Answer: C

Explanation:
The code fragment is using the switch statement with the new Java 17 syntax. The switch statement checks the value of the variable rank and executes the corresponding case statement. In this case, the value of rank is 4, so the first case statement is executed, printing ??Range1??. The second and third case statements are also executed, printing ??Range2?? and ??Range3??. The default case statement is also executed, printing ??Not a valid rank??. References: Java Language Changes - Oracle Help Center

NEW QUESTION 15
Given the code fragment:



```
// Login time:2021-01-12T21:58:18.817Z
Instant loginTime = Instant.now();
Thread.sleep(1000);

// Logout time:2021-01-12T21:58:19.880Z
Instant logoutTime = Instant.now();

loginTime = loginTime.truncatedTo(ChronoUnit.MINUTES);   // line n1
logoutTime = logoutTime.truncatedTo(ChronoUnit.MINUTES);

if (logoutTime.isAfter(loginTime))
    System.out.println("Logged out at: " + logoutTime);
else
    System.out.println("Can't logout");
```

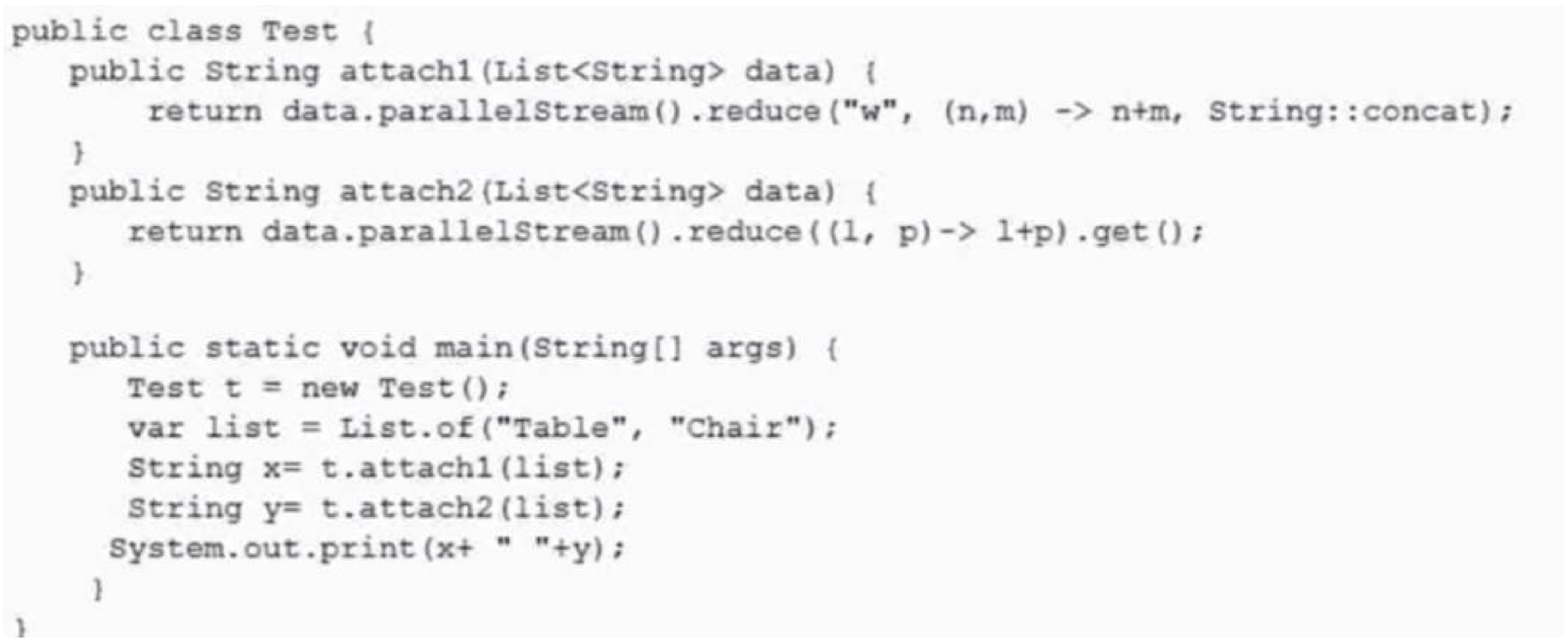
What is the result?

A. Logged out at: 2021-0112T21:58:19.880z
B. Logged out at: 2021-01-12T21:58:00z
C. A compilation error occurs at Line n1.
D. Can??t logout

Answer: B

Explanation:
The code fragment is using the Java SE 17 API to get the current time and then truncating it to minutes. The result will be the current time truncated to minutes, which is why option B is correct. References:
? https://education.oracle.com/products/trackp_OCPJSE17
? https://mylearn.oracle.com/ou/learning-path/java-se-17-developer/99487
? https://docs.oracle.com/javase/17/docs/api/java.base/java/time/Instant.html#truncatedTo(java.time.temporal.TemporalUnit)

NEW QUESTION 20
Given:



```
public class Test {
   public String attach1(List<String> data) {
       return data.parallelStream().reduce("w", (n,m) -> n+m, String::concat);
   }
   public String attach2(List<String> data) {
      return data.parallelStream().reduce((l, p)-> l+p).get();
   }

   public static void main(String[] args) {
      Test t = new Test();
      var list = List.of("Table", "Chair");
      String x= t.attach1(list);
      String y= t.attach2(list);
     System.out.print(x+ " "+y);
    }
}
```

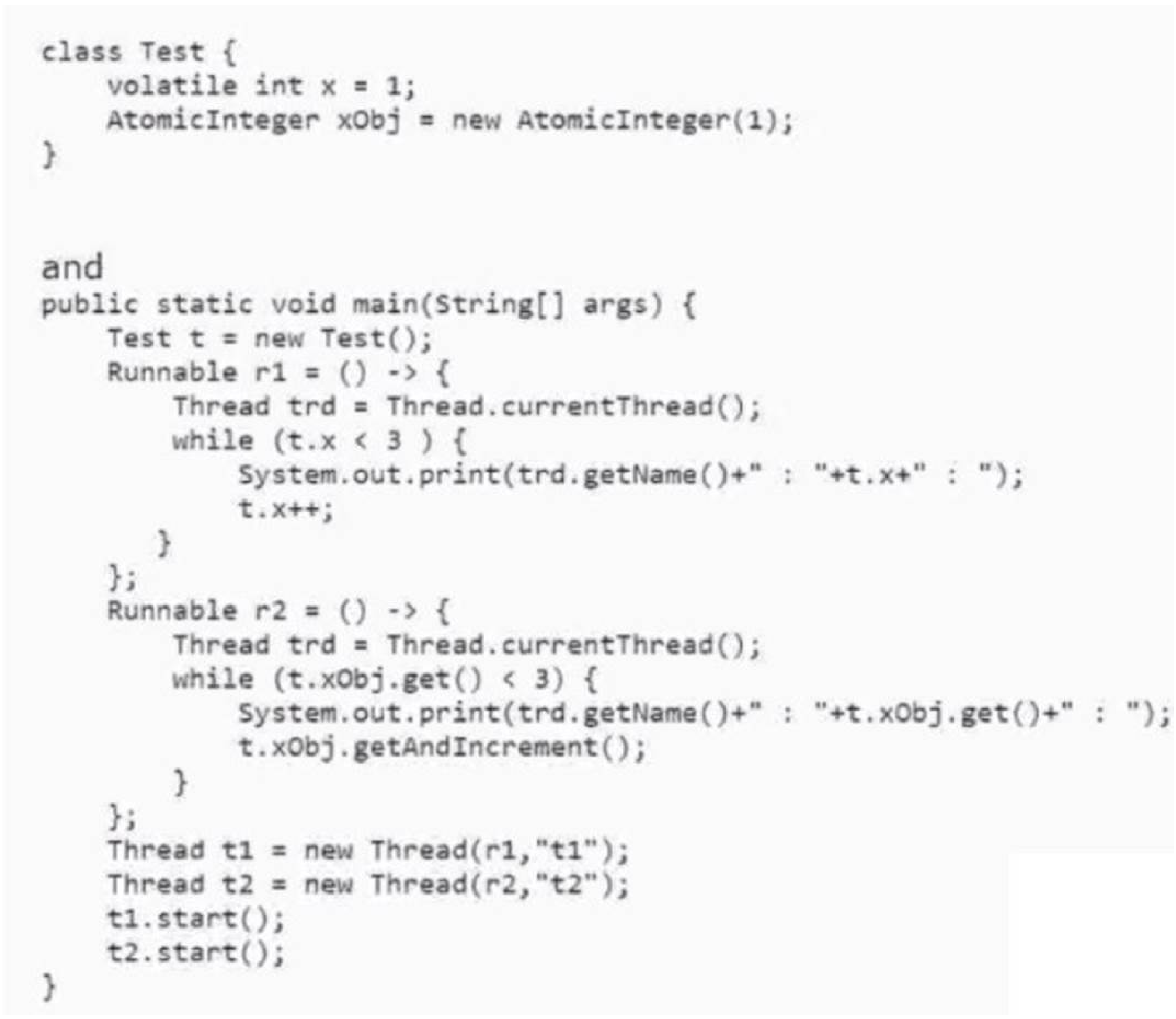
What is the result?

A. Tablechair Tablechair
B. Wtablechair tableChair
C. A RuntimeException is thrown
D. wTableChair TableChair
E. Compilation fails

Answer: E

Explanation:
The code fragment will fail to compile because the class name and the constructor name do not match. The class name is Furniture, but the constructor name is Wtable. This will cause a syntax error. The correct way to define a constructor is to use the same name as the class name. Therefore, the code fragment should change the constructor name to Furniture or change the class name to Wtable.

NEW QUESTION 24
Given the code fragments:



```
class Test {
    volatile int x = 1;
    AtomicInteger xObj = new AtomicInteger(1);
}
```

and

```
public static void main(String[] args) {
    Test t = new Test();
    Runnable r1 = () -> {
        Thread trd = Thread.currentThread();
        while (t.x < 3 ) {
            System.out.print(trd.getName()+" : "+t.x+" : ");
            t.x++;
        }
    };
    Runnable r2 = () -> {
        Thread trd = Thread.currentThread();
        while (t.xObj.get() < 3) {
            System.out.print(trd.getName()+" : "+t.xObj.get()+" : ");
            t.xObj.getAndIncrement();
        }
    };
    Thread t1 = new Thread(r1,"t1");
    Thread t2 = new Thread(r2,"t2");
    t1.start();
    t2.start();
}
```

Which is true?

A. The program prints t1 : 1: t2 : 1: t1 : t2 : 2 : in random order.
B. The program prints t1 : 1 : t2: 1 : t1 : 2 : t2: 2:
C. The program prints t1 : 1: t2 : 1: t1 : 1 : t2 : 1 : indefinitely
D. The program prints an exception

Answer: B

Explanation:
The code creates two threads, t1 and t2, and starts them. The threads will print their names and the value of the Atomic Integer object, x, which is initially set to 1. The threads will then increment the value of x and print their names and the new value of x. Since the threads are started at the same time, the output will be in random order.
However, the final output will always be t1 : 1 : t2: 1 : t1 : 2 : t2: 2: References: AtomicInteger (Java SE 17 & JDK 17) - Oracle

NEW QUESTION 27
......