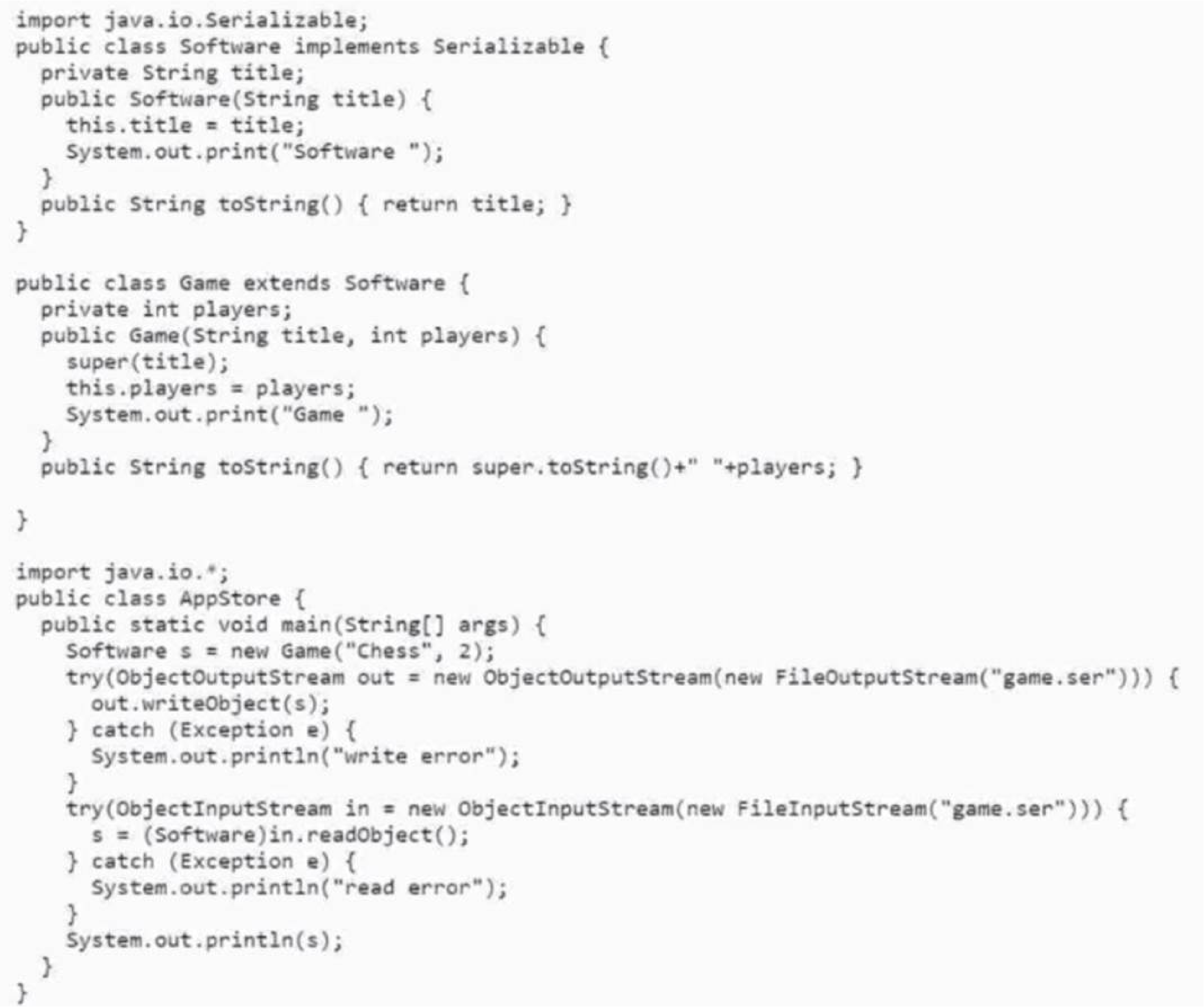
Oracle

Exam Questions 1z0-829

Java SE 17 Developer

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?

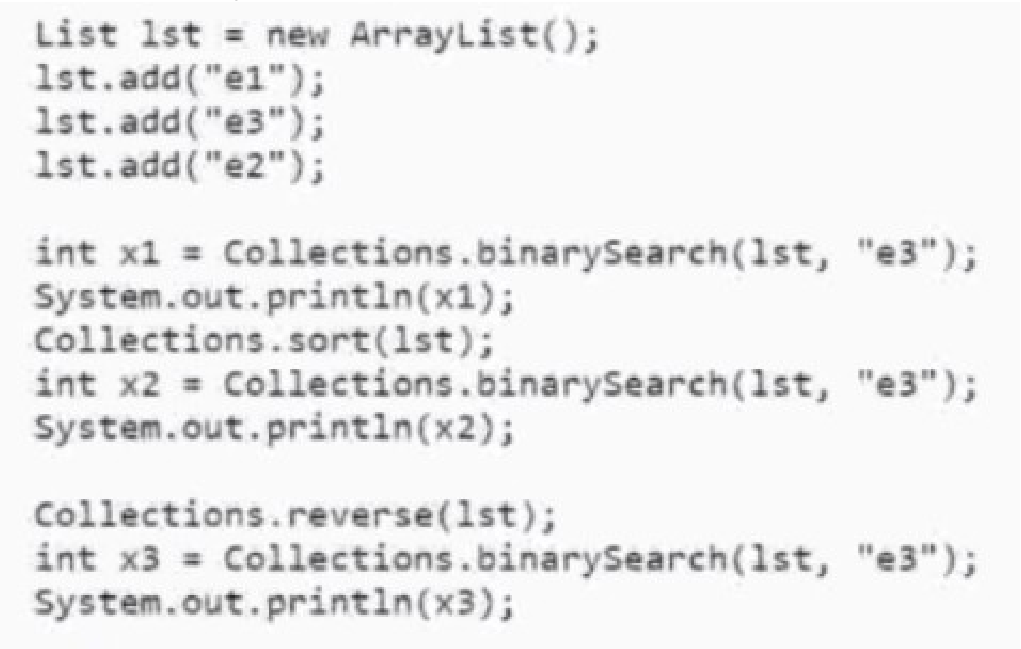
A. IllegalStateException is thrown if a thread in waiting state is moved back to runnable.
B. thread in waiting state consumes CPU cycles.
C. A thread in waiting state must handle InterrupedException.

D. After the timed wait expires, the waited thread moves to the terminated state.

Answer: C

Explanation:
A thread in waiting state is waiting for another thread to perform a particular action, such as calling notify() or notifyAll() on a shared object, or terminating a joined thread. A thread in waiting state can be interrupted by another thread, which will cause the waiting thread to throw an InterruptedException and return to the runnable state. Therefore, a thread in waiting state must handle InterruptedException, either by catching it or declaring it in the throws clause. References: Thread.State (Java SE 17 & JDK 17), [Thread (Java SE 17 & JDK 17)]

NEW QUESTION 3
Given the code fragment:



```
List lst = new ArrayList();
lst.add("e1");
lst.add("e3");
lst.add("e2");

int x1 = Collections.binarySearch(lst, "e3");
System.out.println(x1);
Collections.sort(lst);
int x2 = Collections.binarySearch(lst, "e3");
System.out.println(x2);

Collections.reverse(lst);
int x3 = Collections.binarySearch(lst, "e3");
System.out.println(x3);
```

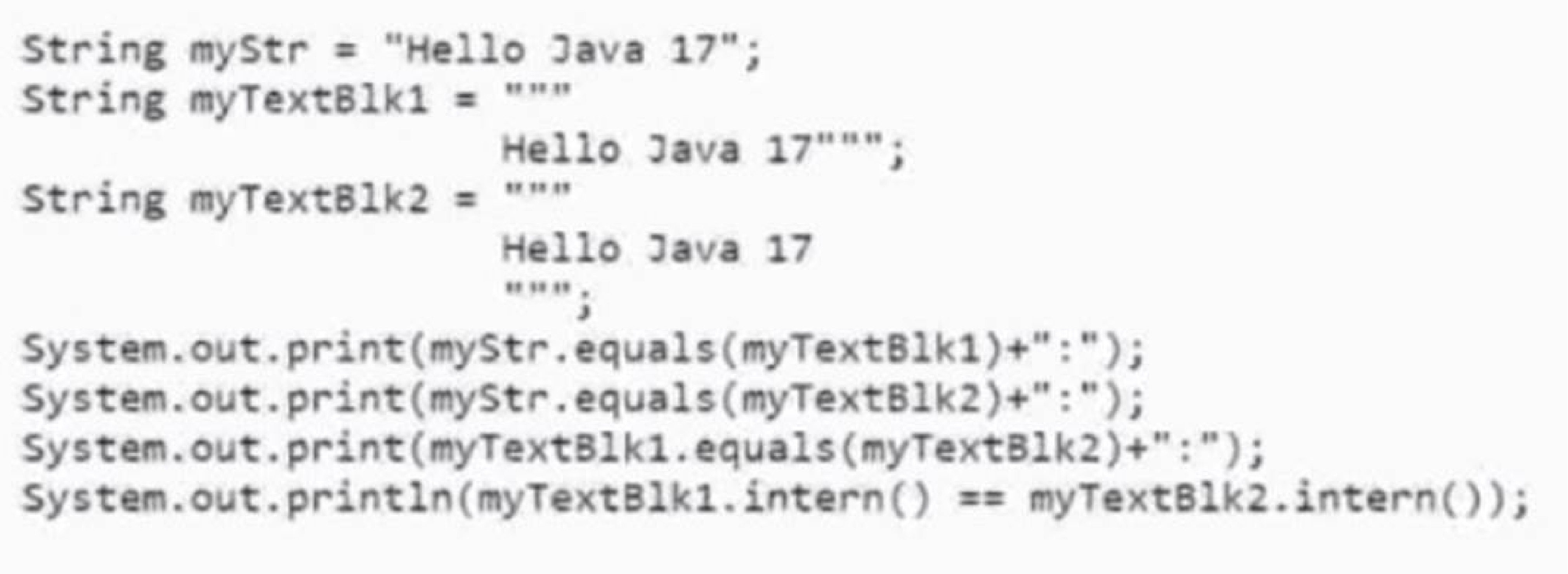
What is the result?

A. 2
B. -2
C. 22E.111F.12-4

Answer: B

Explanation:
The code fragment uses the Collections.binarySearch method to search for the string ??e3?? in the list. The first search returns the index of the element, which is 2. The second search returns the index of the element, which is 0. The third search returns the index of the element, which is -4. The final result is 2. References: Collections (Java SE 17 & JDK 17) - Oracle

NEW QUESTION 4
Given the code fragment:



```
String myStr = "Hello Java 17";
String myTextBlk1 = """
                    Hello Java 17""";
String myTextBlk2 = """
                    Hello Java 17
                    """;
System.out.print(myStr.equals(myTextBlk1)+":");
System.out.print(myStr.equals(myTextBlk2)+":");
System.out.print(myTextBlk1.equals(myTextBlk2)+":");
System.out.println(myTextBlk1.intern() == myTextBlk2.intern());
```

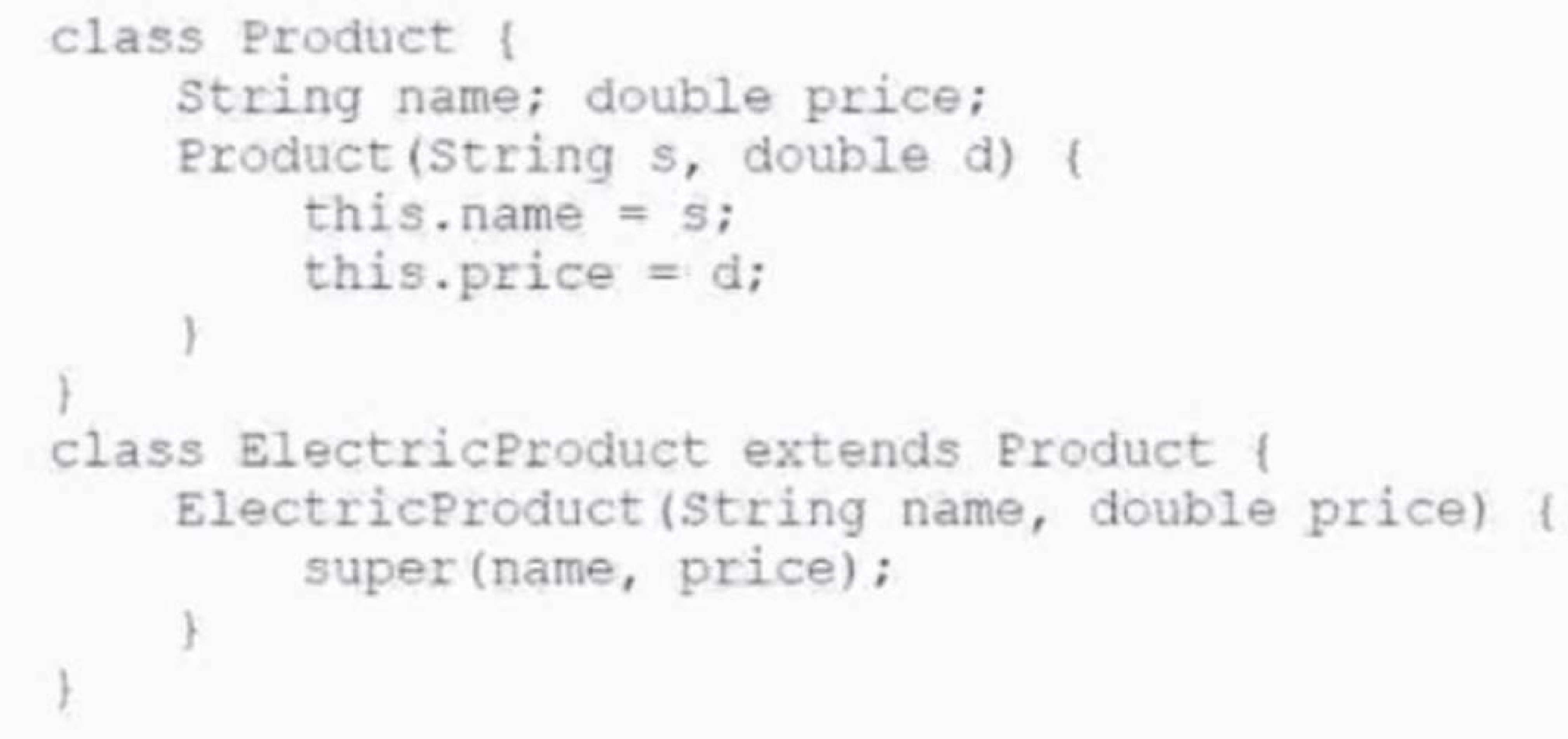
A. True:false:true:true
B. True:true:false:false
C. True:false:true:false
D. True:false:false:false

Answer: C

Explanation:
The code fragment compares four pairs of strings using the equals() and intern() methods. The equals() method compares the content of two strings, while the intern() method returns a canonical representation of a string, which means that it returns a reference to an existing string with the same content in the string pool. The string pool is a memory area where strings are stored and reused to save space and improve performance. The results of the comparisons are as follows:
? s1.equals(s2): This returns true because both s1 and s2 have the same content, ??Hello Java 17??.

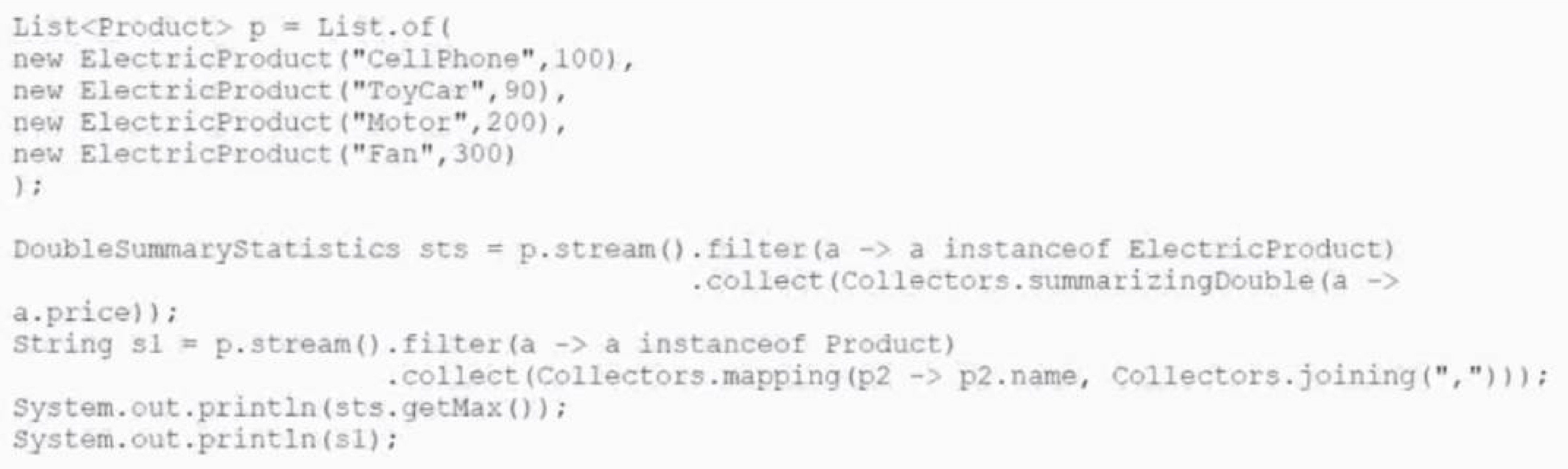
? s1 == s2: This returns false because s1 and s2 are different objects with different references, even though they have the same content. The == operator compares the references of two objects, not their content.
? s1.intern() == s2.intern(): This returns true because both s1.intern() and s2.intern() return a reference to the same string object in the string pool, which has the content ??Hello Java 17??. The intern() method ensures that there is only one copy of each distinct string value in the string pool.
? ??Hello Java 17?? == s2: This returns false because ??Hello Java 17?? is a string literal, which is automatically interned and stored in the string pool, while s2 is a string object created with the new operator, which is not interned by default and stored in the heap. Therefore, they have different references and are not equal using the == operator.
References: String (Java SE 17 & JDK 17) - Oracle

NEW QUESTION 5
Given:



```
class Product {
    String name; double price;
    Product(String s, double d) {
        this.name = s;
        this.price = d;
    }
}
class ElectricProduct extends Product {
    ElectricProduct(String name, double price) {
        super(name, price);
    }
}
```

and the code fragment:



```
List<Product> p = List.of(
new ElectricProduct("CellPhone",100),
new ElectricProduct("ToyCar",90),
new ElectricProduct("Motor",200),
new ElectricProduct("Fan",300)
);

DoubleSummaryStatistics sts = p.stream().filter(a -> a instanceof ElectricProduct)
                                        .collect(Collectors.summarizingDouble(a ->
a.price));
String s1 = p.stream().filter(a -> a instanceof Product)
                      .collect(Collectors.mapping(p2 -> p2.name, Collectors.joining(",")));
System.out.println(sts.getMax());
System.out.println(s1);
```

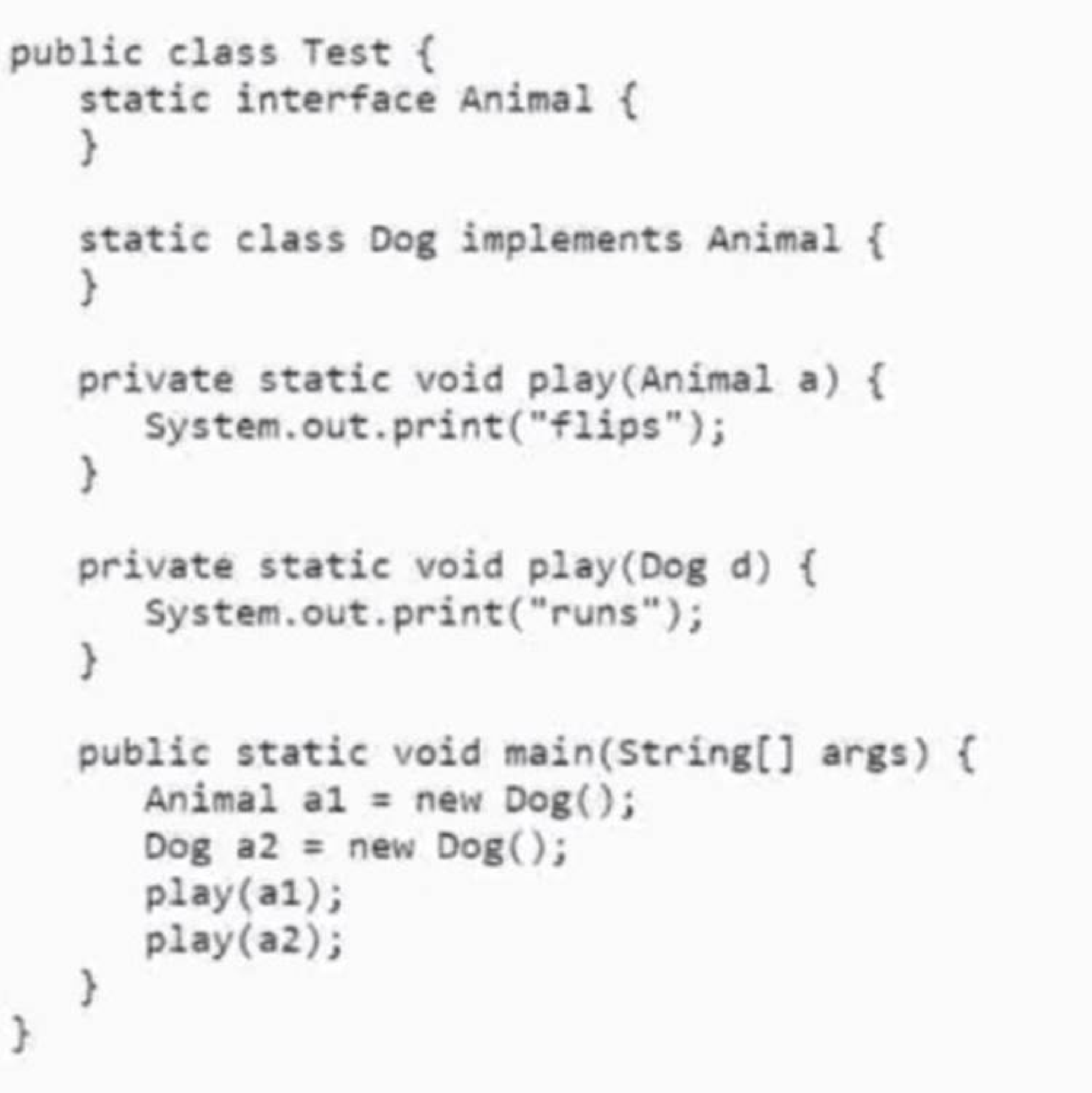
A. 300.00CellPhone,ToyCar,Motor,Fan
B. 100.00CellPhone,ToyCar,Motor,Fan
C. 100.00 CellPhone,ToyCar
D. 300.00CellPhone.ToyCar

Answer: A

Explanation:
The code fragment is using the Stream API to perform a reduction operation on a list of ElectricProduct objects. The reduction operation consists of three parts: an identity value, an accumulator function, and a combiner function. The identity value is the initial value of the result, which is 0.0 in this case. The accumulator function is a BiFunction that takes two arguments: the current result and the current element of the stream, and returns a new result. In this case, the accumulator function is (a,b) -> a + b.getPrice (), which means that it adds the price of each element to the current result. The combiner function is a BinaryOperator that takes two partial results and combines them into one. In this case, the combiner function is (a,b) -> a + b, which means that it adds the two partial results together.
The code fragment then applies a filter operation on the stream, which returns a new stream that contains only the elements that match the given predicate. The predicate is p -
> p.getPrice () > 10, which means that it selects only the elements that have a price greater than 10. The code fragment then applies a map operation on the filtered stream, which returns a new stream that contains the results of applying the given function to each element. The function is p -> p.getName (), which means that it returns the name of each element.
The code fragment then calls the collect method on the mapped stream, which performs a mutable reduction operation on the elements of the stream using a Collector. The Collector is Collectors.joining (??,??), which means that it concatenates the elements of the stream into a single String, separated by commas.
The code fragment then prints out the result of the reduction operation and the result of the collect operation, separated by a new line. The result of the reduction operation is 300.00, which is the sum of the prices of all ElectricProduct objects that have a price greater than 10. The result of the collect operation is CellPhone,ToyCar,Motor,Fan, which is the concatenation of the names of all ElectricProduct objects that have a price greater than 10. Therefore, the output of the

code fragment is: 300.00 CellPhone,ToyCar,Motor,Fan
References: Stream (Java SE 17 & JDK 17) - Oracle, Collectors (Java SE 17 & JDK 17) - Oracle

NEW QUESTION 6
Given:



```
public class Test {
    static interface Animal {
    }

    static class Dog implements Animal {
    }

    private static void play(Animal a) {
        System.out.print("flips");
    }

    private static void play(Dog d) {
        System.out.print("runs");
    }

    public static void main(String[] args) {
        Animal a1 = new Dog();
        Dog a2 = new Dog();
        play(a1);
        play(a2);
    }
}
```

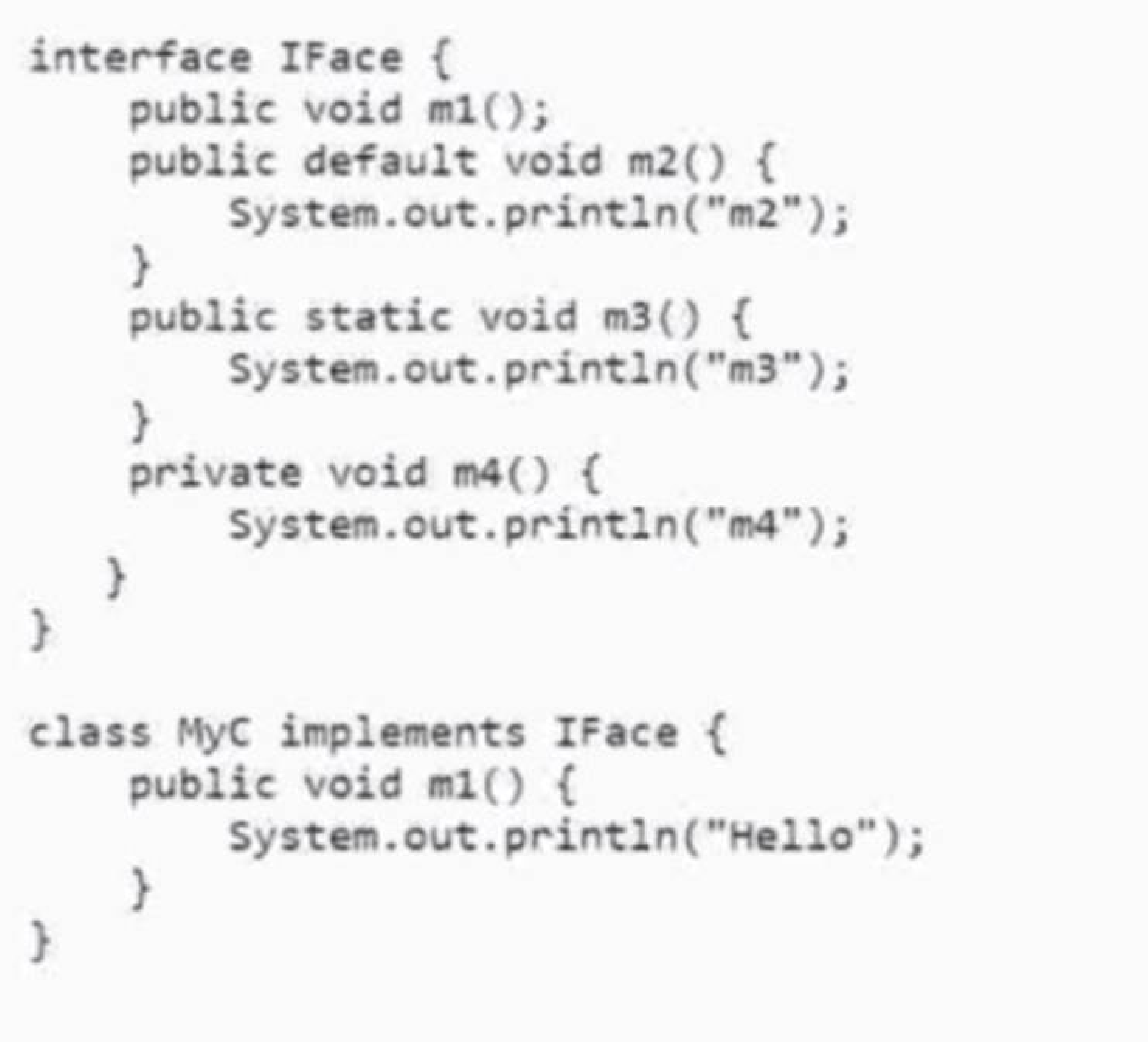
What is the result?

A. flipsflips
B. Compilation fails
C. flipsruns
D. runsflips
E. runsruns

Answer: B

Explanation:
The code fragment will fail to compile because the play method in the Dog class is declared as private, which means that it cannot be accessed from outside the class. The main method is trying to call the play method on a Dog object, which is not allowed. Therefore, the code fragment will produce a compilation error.

NEW QUESTION 7
Given:



```
interface IFace {
    public void m1();
    public default void m2() {
        System.out.println("m2");
    }
    public static void m3() {
        System.out.println("m3");
    }
    private void m4() {
        System.out.println("m4");
   }
}

class MyC implements IFace {
    public void m1() {
        System.out.println("Hello");
    }
}
```

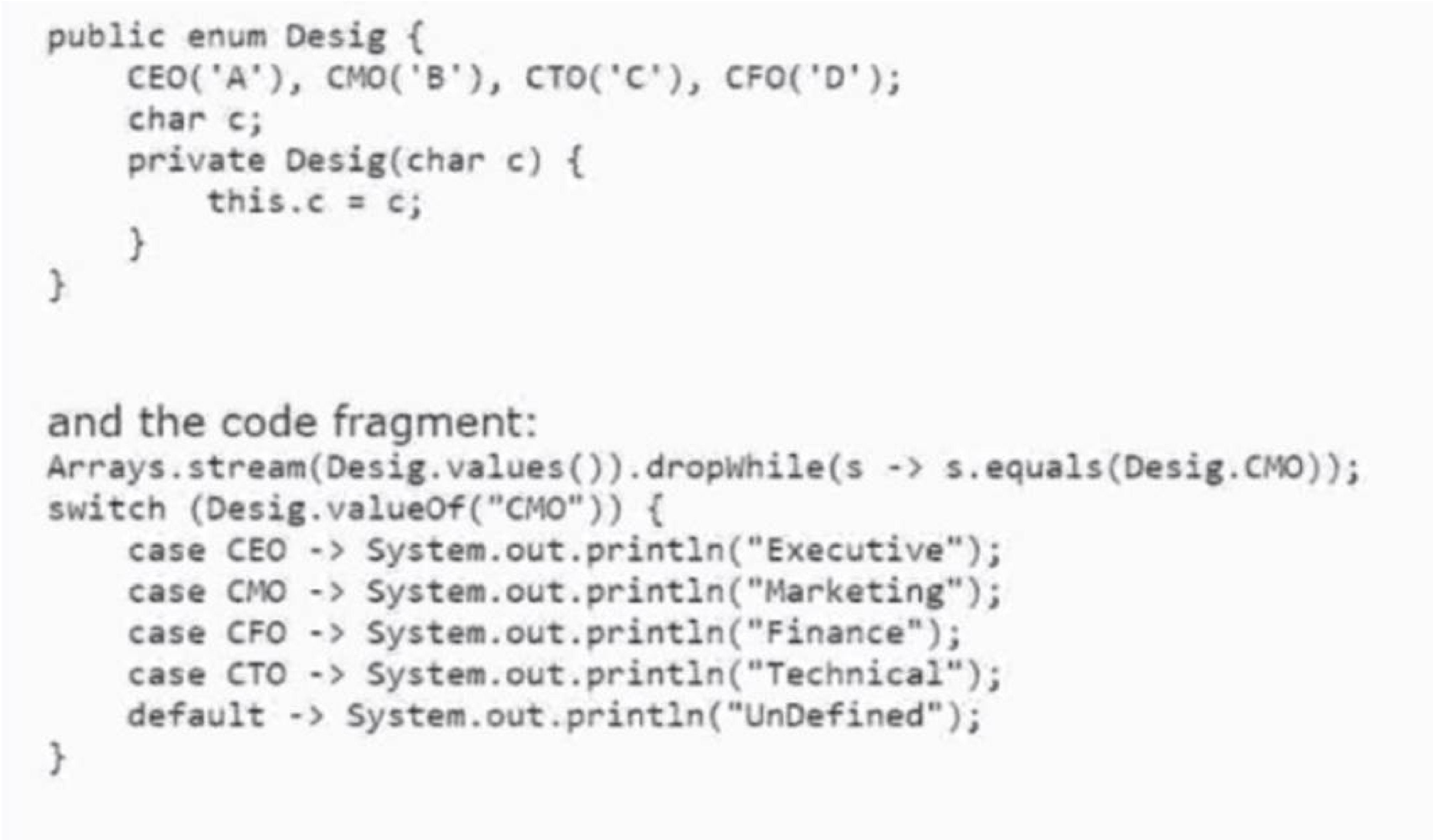
Which two method invocation execute?

A. IFace myclassobj = new Myc (); myclassObj.m3 ();
B. Ifnce.m3 ();
C. iFace mucloassObj = new Myc (); myClassObj.m4();
D. new MyC() .m2 ();
E. IFace .,4():
F. IFace.m2();

Answer: DE

Explanation:
The code given is an interface and a class that implements the interface. The interface has three methods, m1(), m2(), and m3(). The class has one method, m1(). The only two method invocations that will execute are D and E. D is a call to the m2() method in the class, and E is a call to the m3() method in the interface.
References: https://education.oracle.com/products/trackp_OCPJSE17, 3, 4, 5

NEW QUESTION 8
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");
}
```

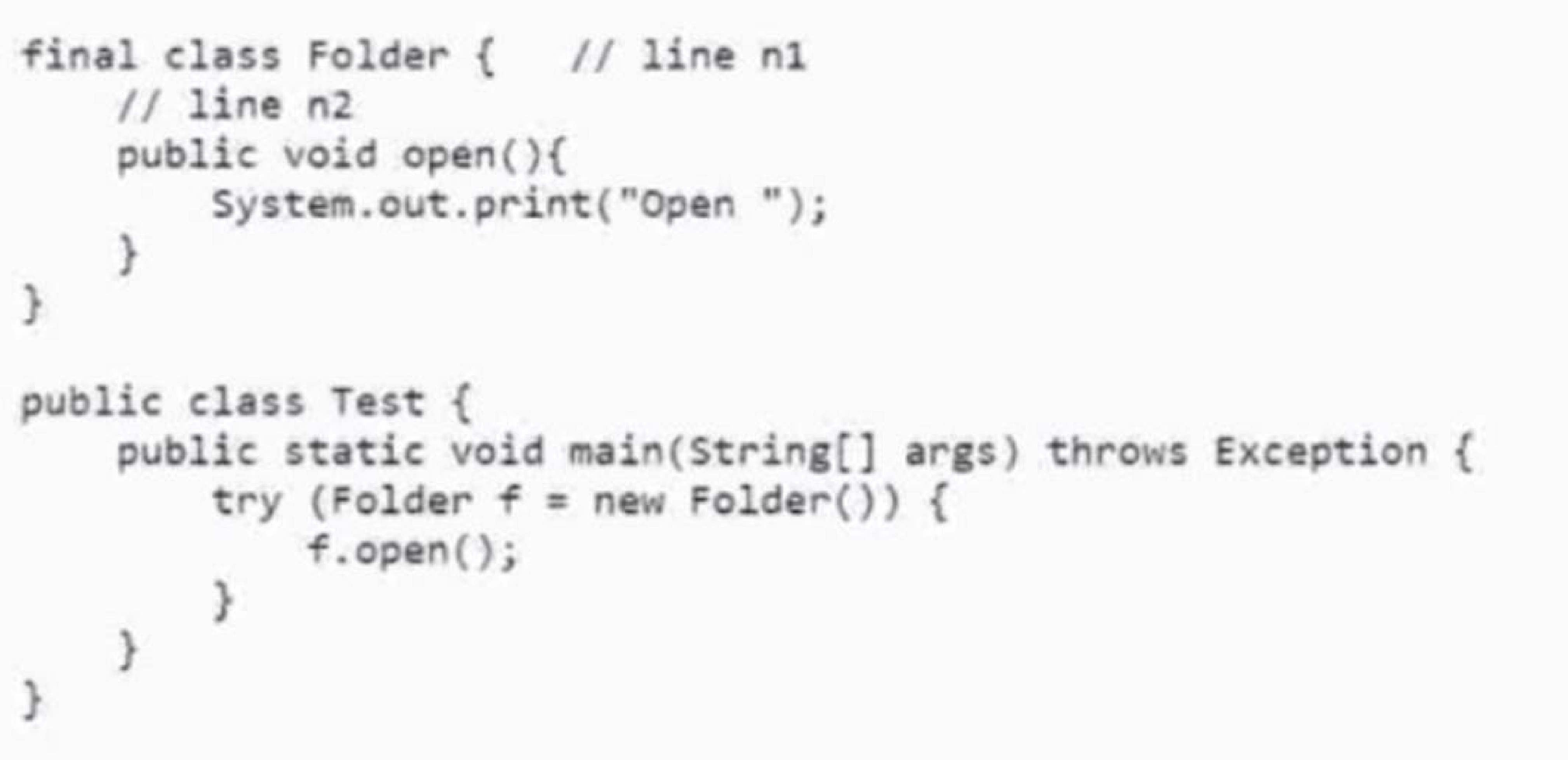
What is the result

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

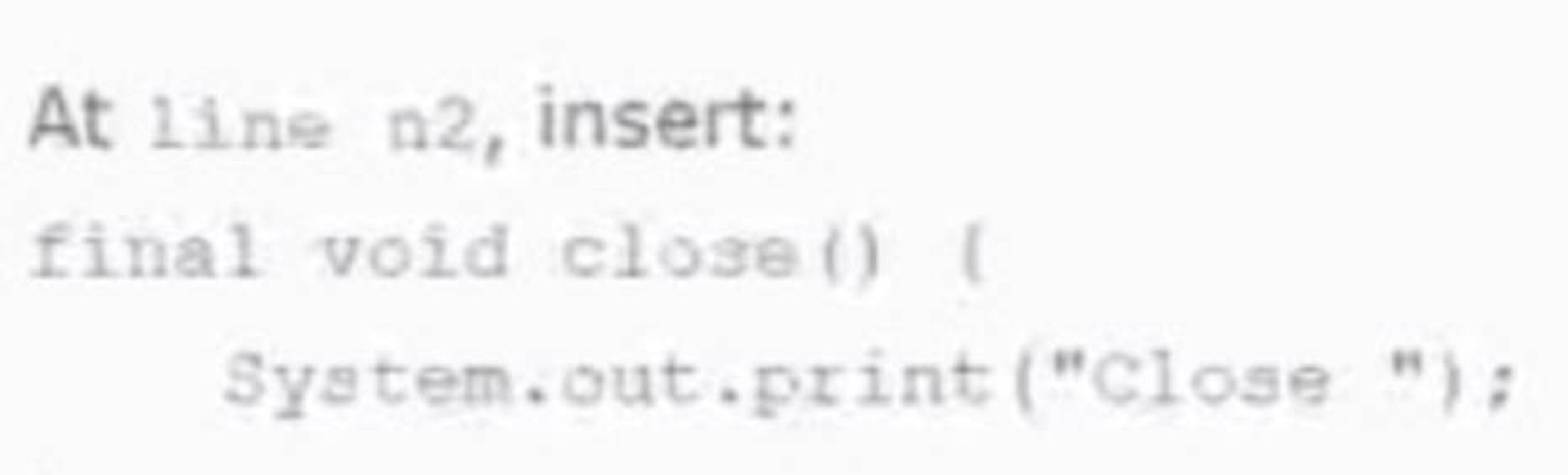
NEW QUESTION 9
Given:



```
final class Folder {    // line n1
    // line n2
    public void open(){
        System.out.print("Open ");
    }
}

public class Test {
    public static void main(String[] args) throws Exception {
        try (Folder f = new Folder()) {
            f.open();
        }
    }
}
```

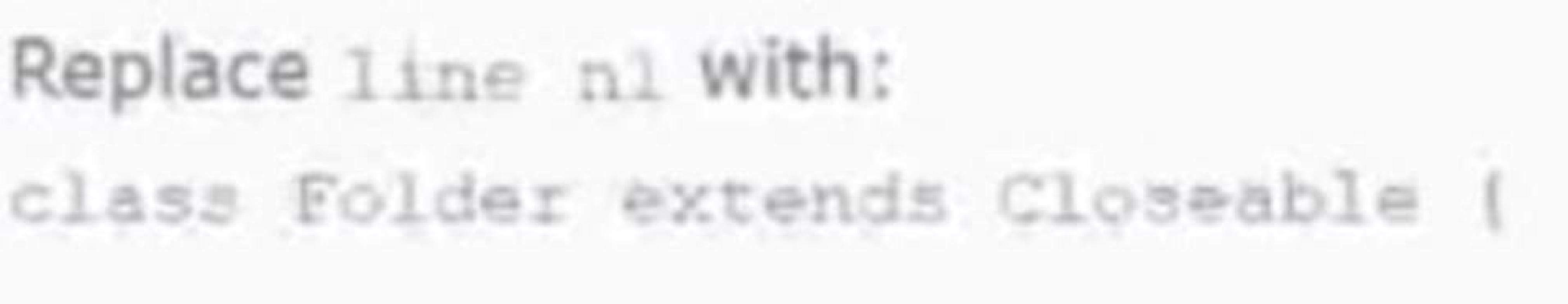
Which two modifications enable the code to print Open Close?
A)



At line n2, insert:

```
final void close() {
      System.out.print("Close ");
```

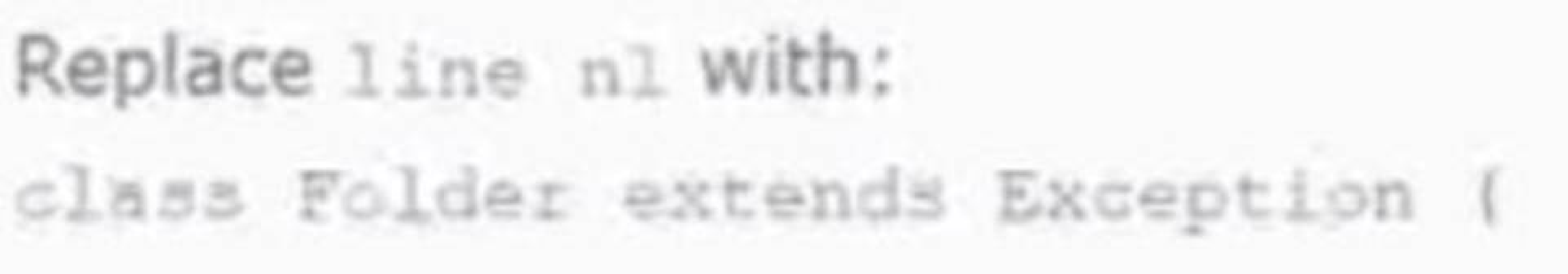
B)



Replace line n1 with:

```
class Folder extends Closeable {
```

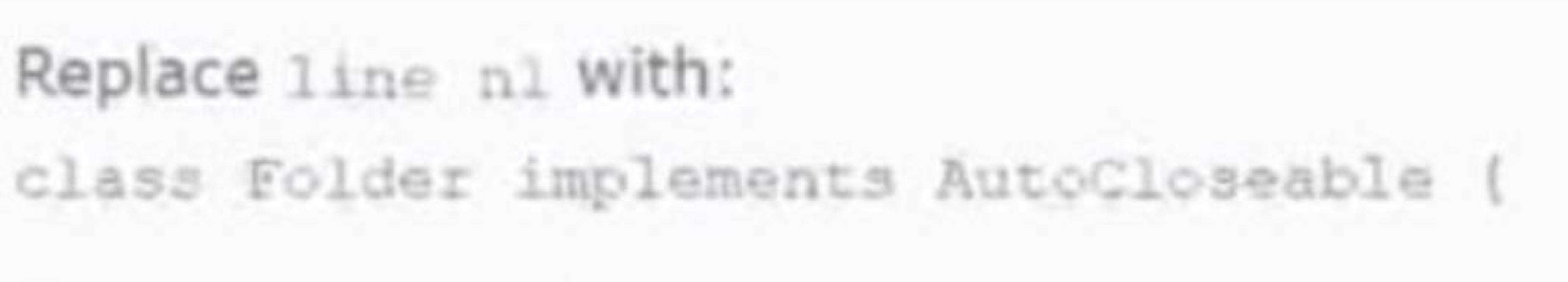
C)



Replace line n1 with:

```
class Folder extends Exception {
```

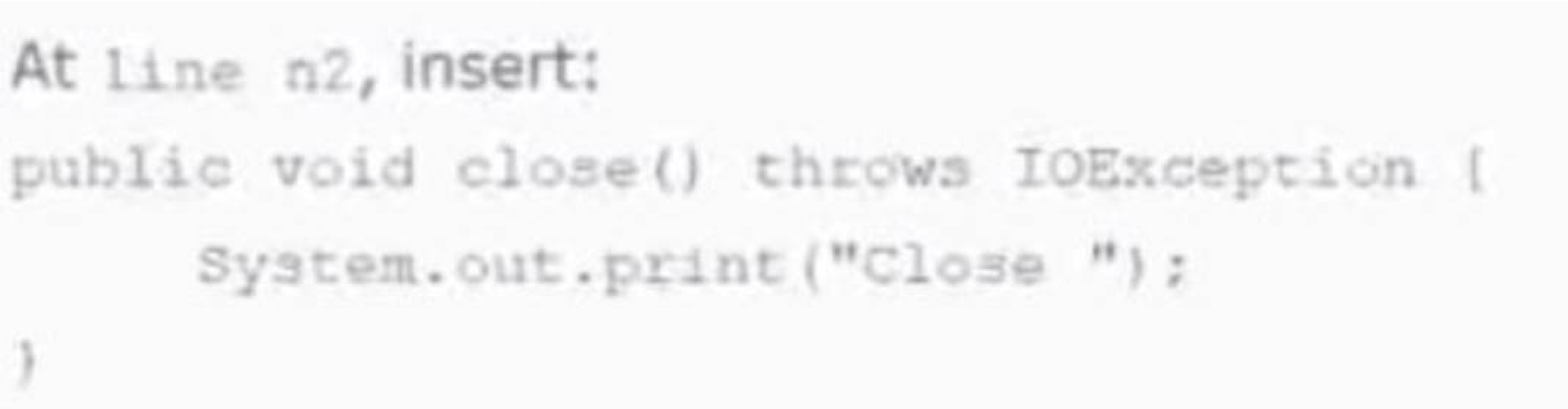
D)



Replace line n1 with:

```
class Folder implements AutoCloseable {
```

E)



At line n2, insert:

```
public void close() throws IOException {
      System.out.print("Close ");
}
```

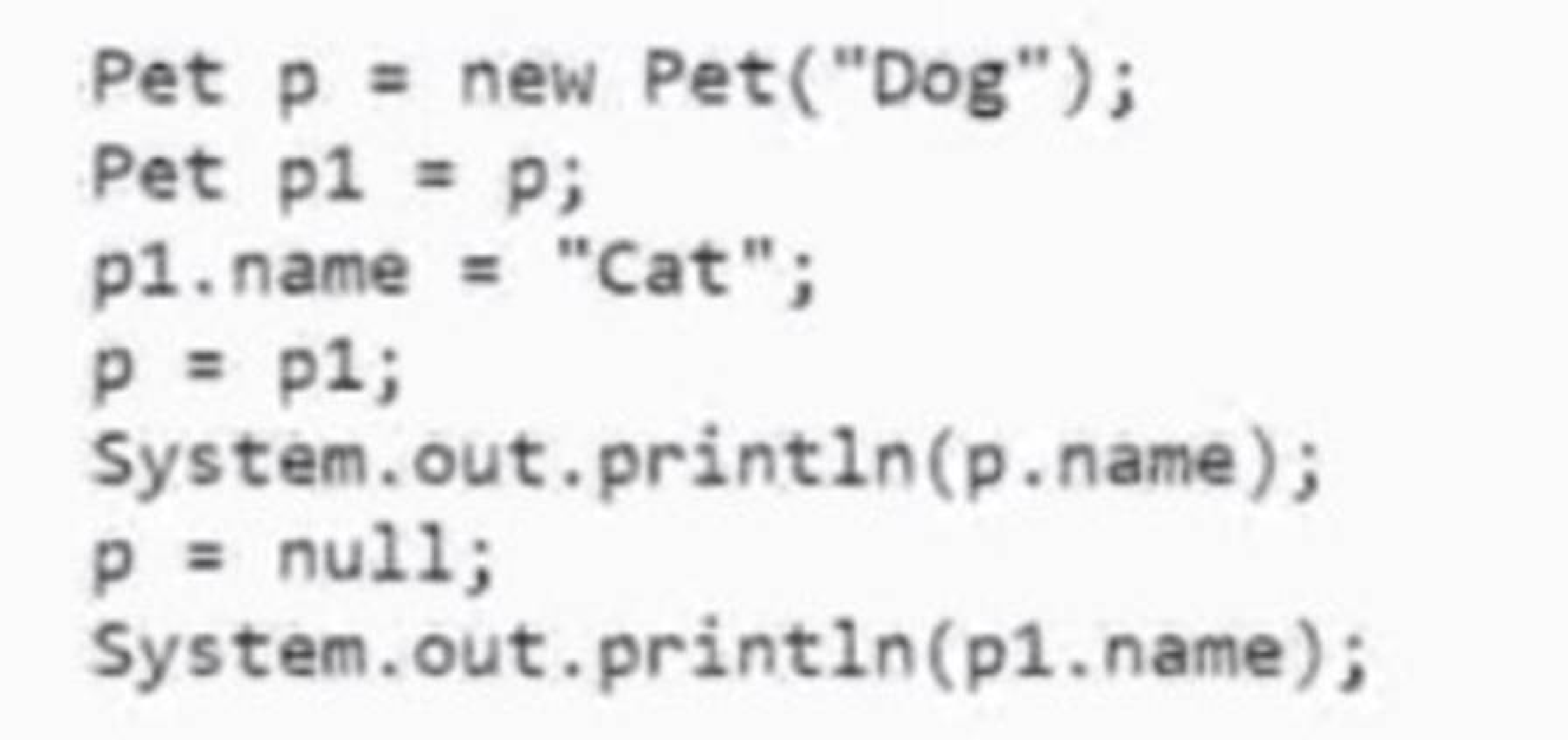
A. Option A
B. Option B
C. Option C
D. Option D
E. Option E

Answer: BE

Explanation:
The code given is a try-with-resources statement that declares a resource of type AutoCloseable. The resource is an anonymous class that implements the AutoCloseable interface and overrides the close() method. The code also has a print() method that prints the value of the variable s. The code is supposed to print ??Open Close??, but it does not compile because of two errors.

The first error is at line n1, where the anonymous class is missing a semicolon at the end of its declaration. This causes a syntax error and prevents the code from compiling. To fix this error, option B adds a semicolon after the closing curly brace of the anonymous class.
The second error is at line n2, where the print() method is called without an object reference. This causes a compilation error because the print() method is not static and cannot be invoked without an object. To fix this error, option E adds an object reference to the print() method by using the variable t.
Therefore, options B and E are correct and enable the code to print ??Open Close??.

NEW QUESTION 10
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);
```

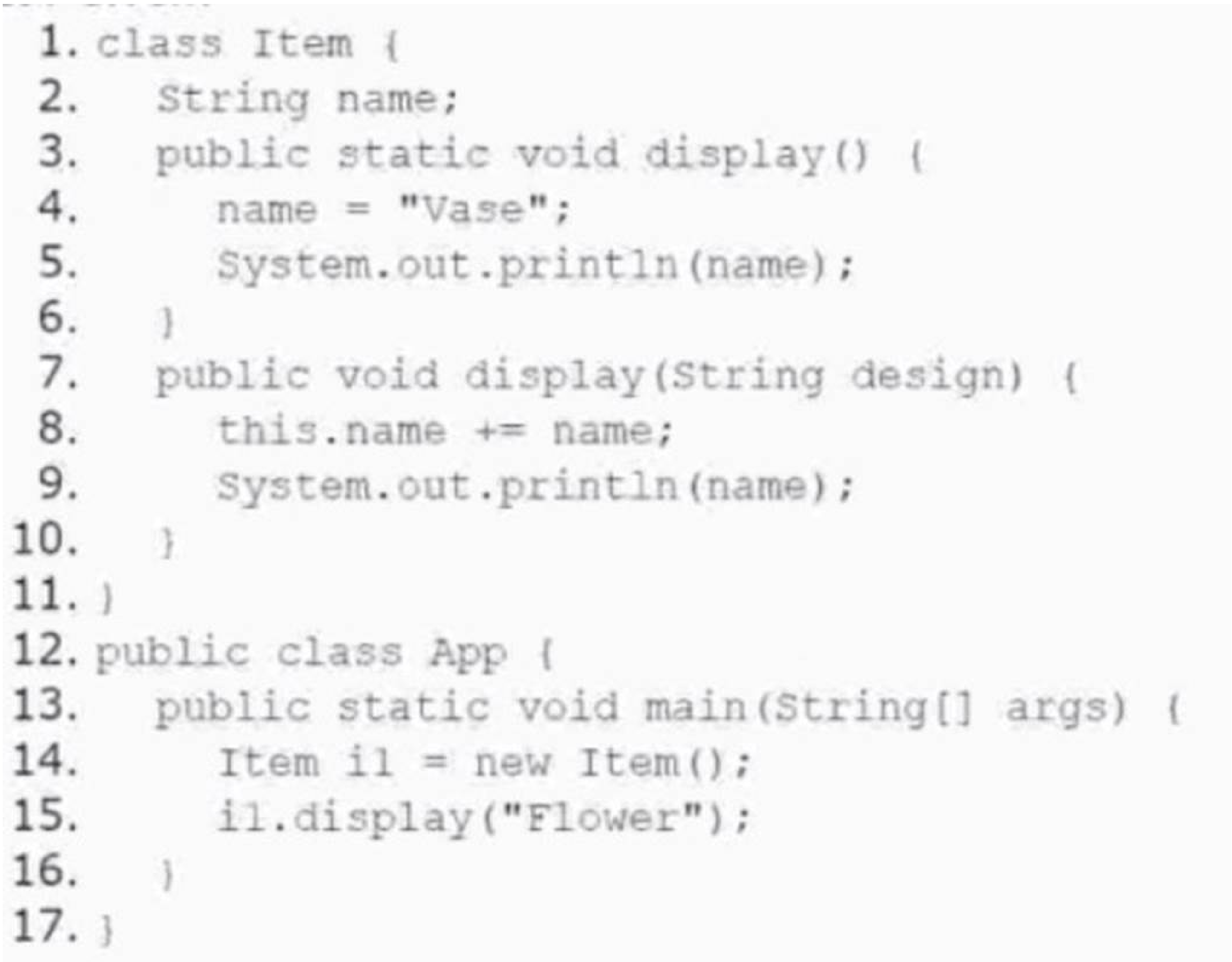
What is the result?

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 10
Given:



```
 1. class Item {
 2.     String name;
 3.     public static void display() {
 4.         name = "Vase";
 5.         System.out.println(name);
 6.     }
 7.     public void display(String design) {
 8.         this.name += name;
 9.         System.out.println(name);
10.     }
11. }
12. public class App {
13.     public static void main(String[] args) {
14.         Item i1 = new Item();
15.         i1.display("Flower");
16.     }
17. }
```

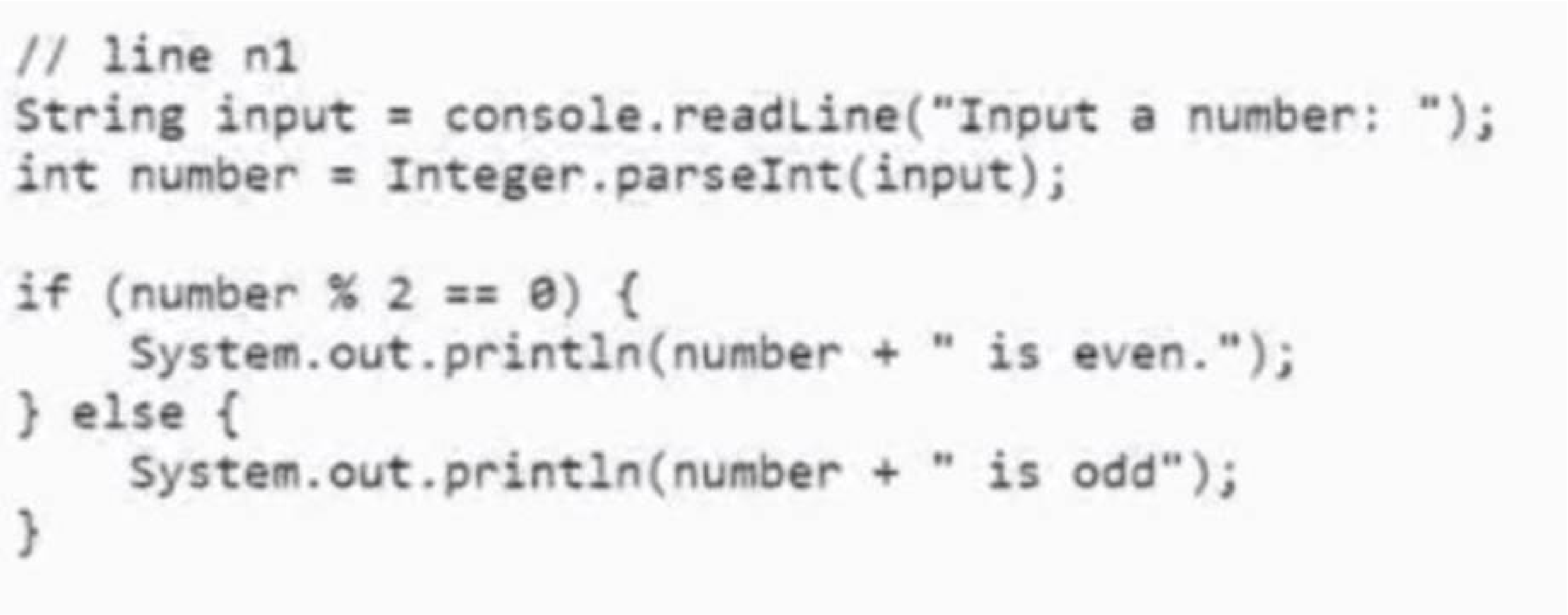
Which action enables the code to compile?

A. Replace 15 with item.display ("Flower");
B. Replace 2 with static string name;
C. Replace 7 with public void display (string design) {
D. Replace 3 with private static void display () {

Answer: C

Explanation:
The answer is C because the code fragment contains a syntax error in line 7, where the method display is declared without any parameter type. This causes a compilation error, as Java requires the parameter type to be specified for each method parameter. To fix this error, the parameter type should be added before the parameter name, such as string design. This will enable the code to compile and run without any errors. References:
? Oracle Certified Professional: Java SE 17 Developer
? Java SE 17 Developer
? OCP Oracle Certified Professional Java SE 17 Developer Study Guide
? Java Methods

NEW QUESTION 14
Given the code fragment:

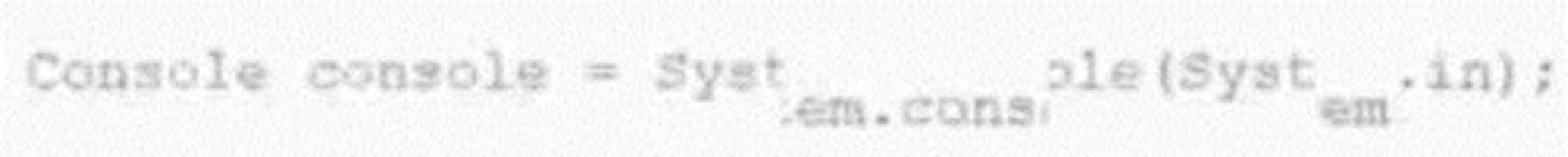


```
// line n1
String input = console.readLine("Input a number: ");
int number = Integer.parseInt(input);

if (number % 2 == 0) {
    System.out.println(number + " is even.");
} else {
    System.out.println(number + " is odd");
}
```

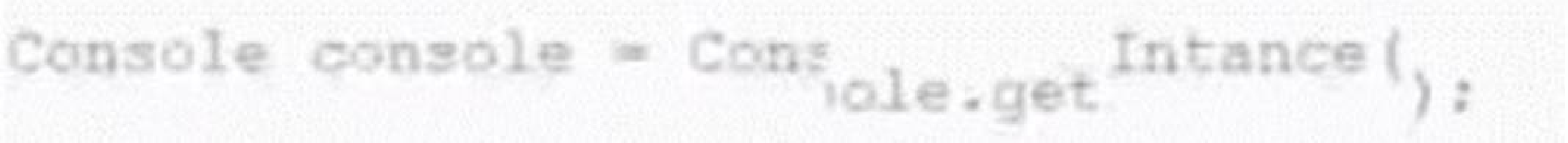
Which code line n1, obtains the java.io.Console object?

A)



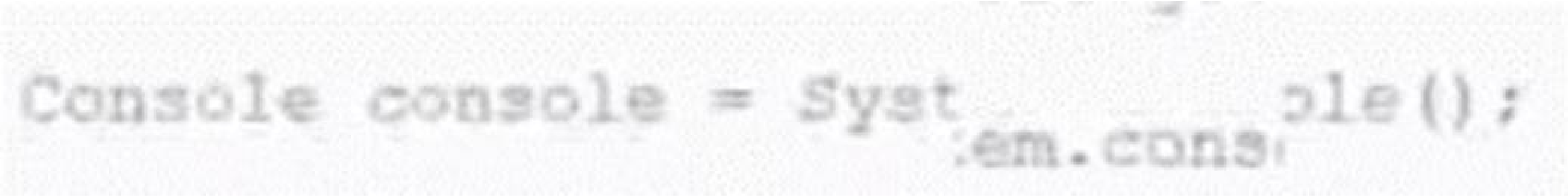
```
Console console = System.console(System.in);
```

B)



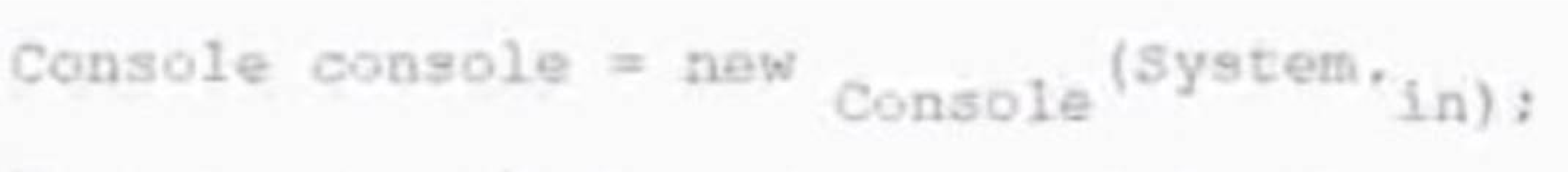
```
Console console = Console.getIntance();
```

C)



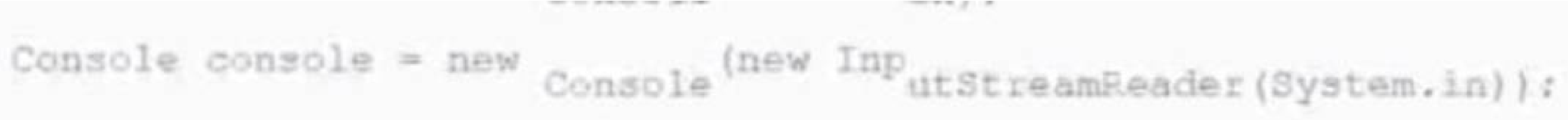
```
Console console = System.console();
```

D)



```
Console console = new Console(System.in);
```

E)



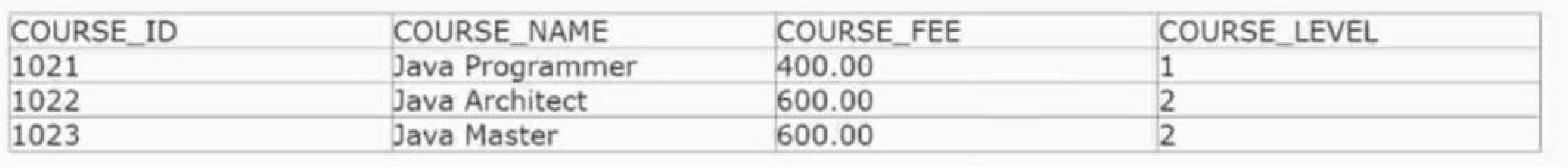
```
Console console = new Console(new InputStreamReader(System.in));
```

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

Answer: A

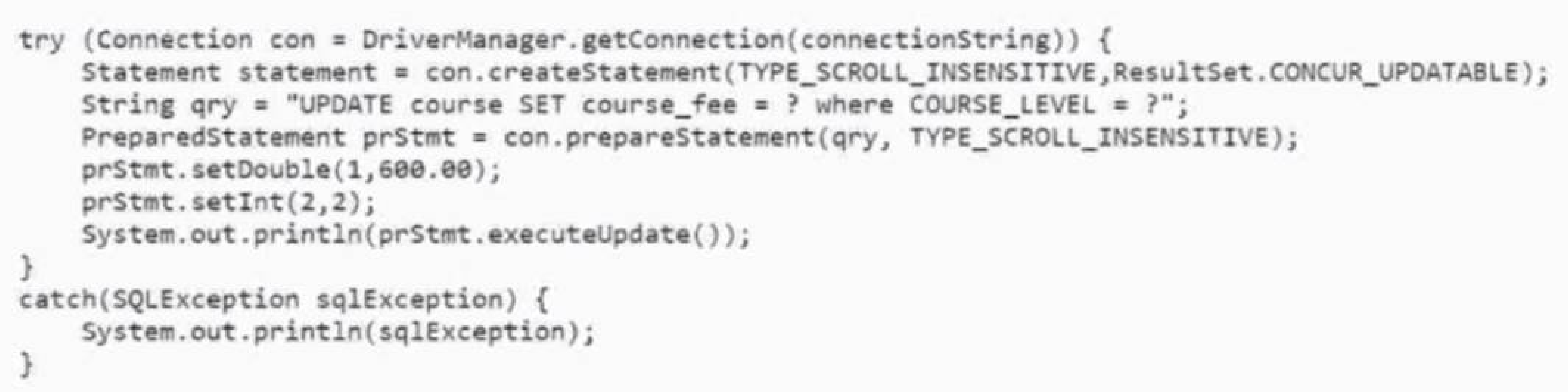
Explanation:
The code fragment is trying to obtain the java.io.Console object, which is a class that provides methods to access the character-based console device, if any, associated with the current Java virtual machine. The correct way to obtain the Console object is to call the static method Console console() in the java.lang.System class. This method returns the unique Console object associated with the current Java virtual machine, if any. Therefore, option A is correct, as it calls System.console() and assigns it to a Console variable. References:
? https://docs.oracle.com/javase/17/docs/api/java.base/java/io/Console.html
? https://docs.oracle.com/javase/17/docs/api/java.base/java/lang/System.html#cons ole()
? https://education.oracle.com/products/trackp_OCPJSE17
? https://mylearn.oracle.com/ou/learning-path/java-se-17-developer/99487

NEW QUESTION 17
Given the course table:



COURSE_ID	COURSE_NAME	COURSE_FEE	COURSE_LEVEL
1021	Java Programmer	400.00	1
1022	Java Architect	600.00	2
1023	Java Master	600.00	2

Given the code fragment:



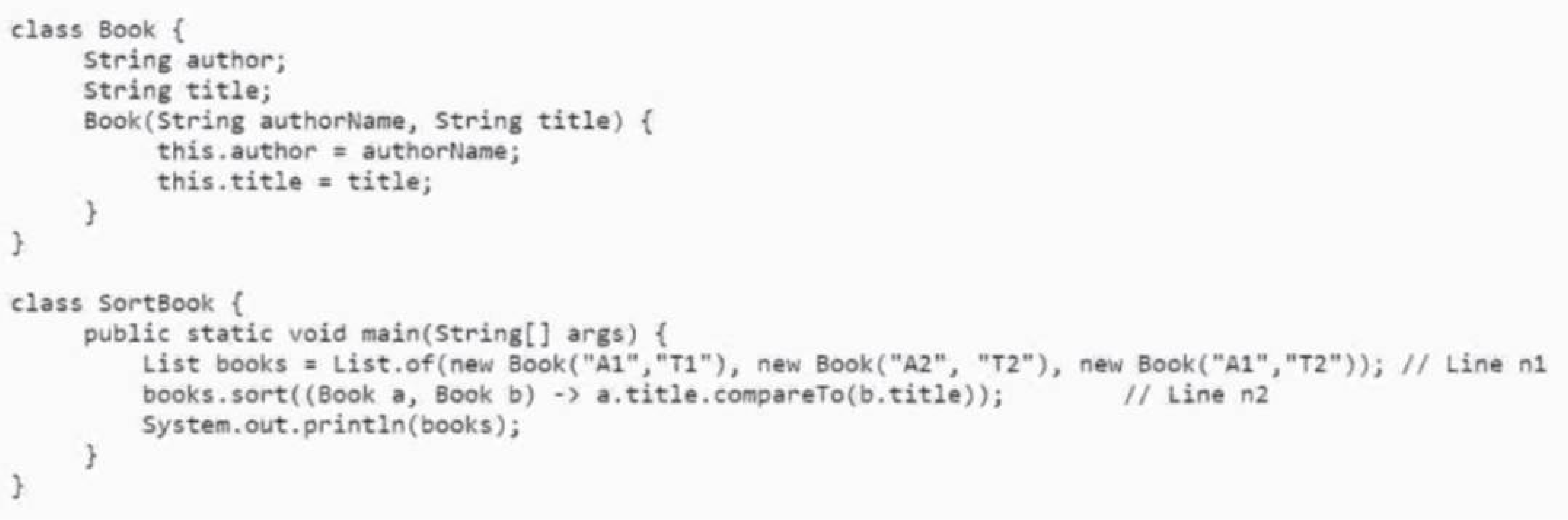
```
try (Connection con = DriverManager.getConnection(connectionString)) {
    Statement statement = con.createStatement(TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_UPDATABLE);
    String qry = "UPDATE course SET course_fee = ? where COURSE_LEVEL = ?";
    PreparedStatement prStmt = con.prepareStatement(qry, TYPE_SCROLL_INSENSITIVE);
    prStmt.setDouble(1,600.00);
    prStmt.setInt(2,2);
    System.out.println(prStmt.executeUpdate());
}
catch(SQLException sqlException) {
    System.out.println(sqlException);
}
```

A. 2
B. false
C. true
D. 1

Answer: C

Explanation:
The code fragment will execute the update statement and set the course fee of the course with ID 1021 to 5000. The executeUpdate method returns an int value that indicates the number of rows affected by the SQL statement. In this case, only one row will be updated, so the result variable will be 1. The if statement will check if the result is greater than 0, which is true, and print ??Updated successfully??. Therefore, the output of the code fragment is true. References: https://education.oracle.com/products/trackp_OCPJSE17, https://mylearn.oracle.com/ou/learning-path/java-se-17-developer/99487, https://docs.oracle.com/en/java/javase/17/docs/api/java.sql/java/sql/Statement.html#execut eUpdate(java.lang.String)

NEW QUESTION 20
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);
    }
}
```

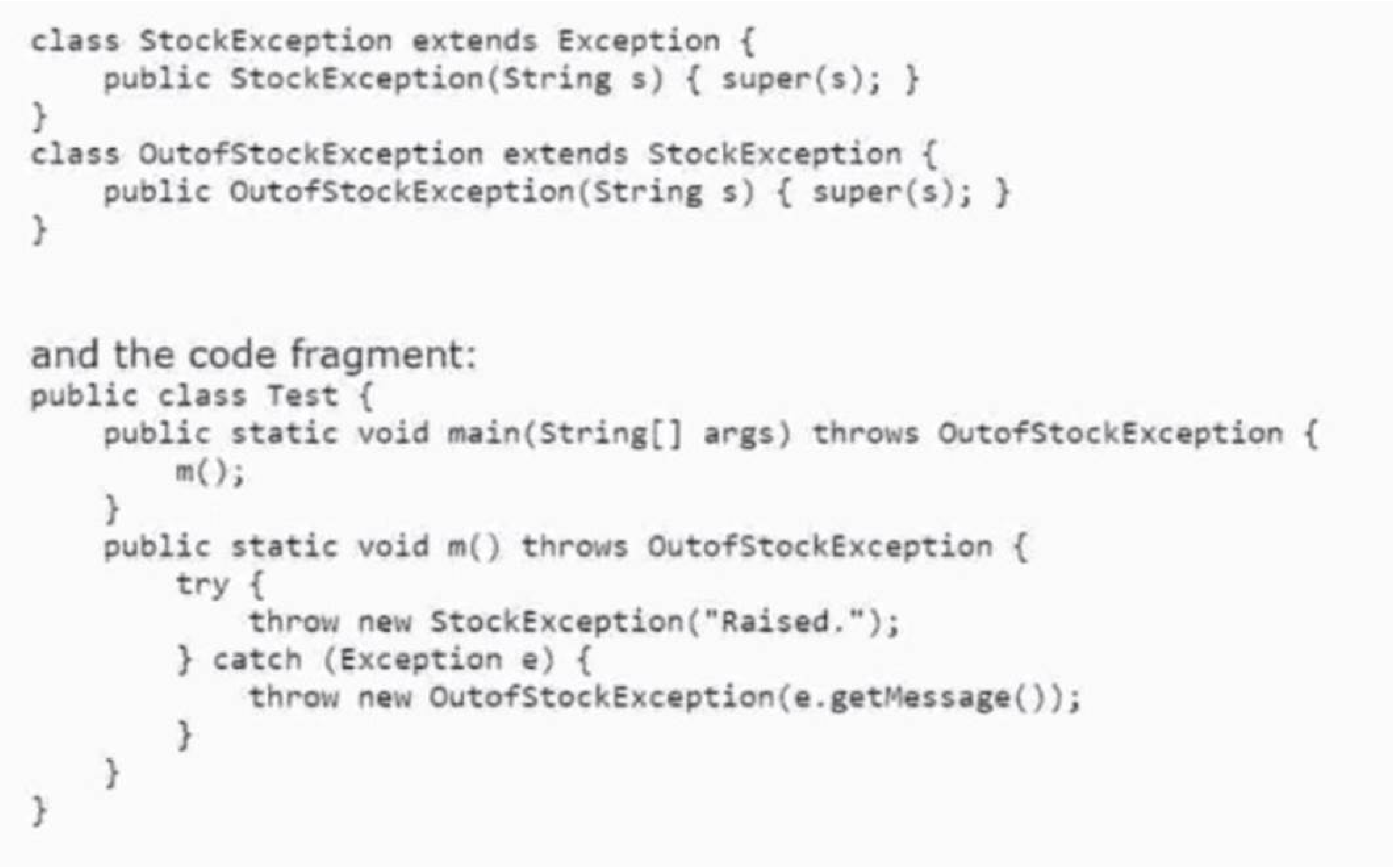
Which action sorts the book list?

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 24
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());
        }
    }
}
```

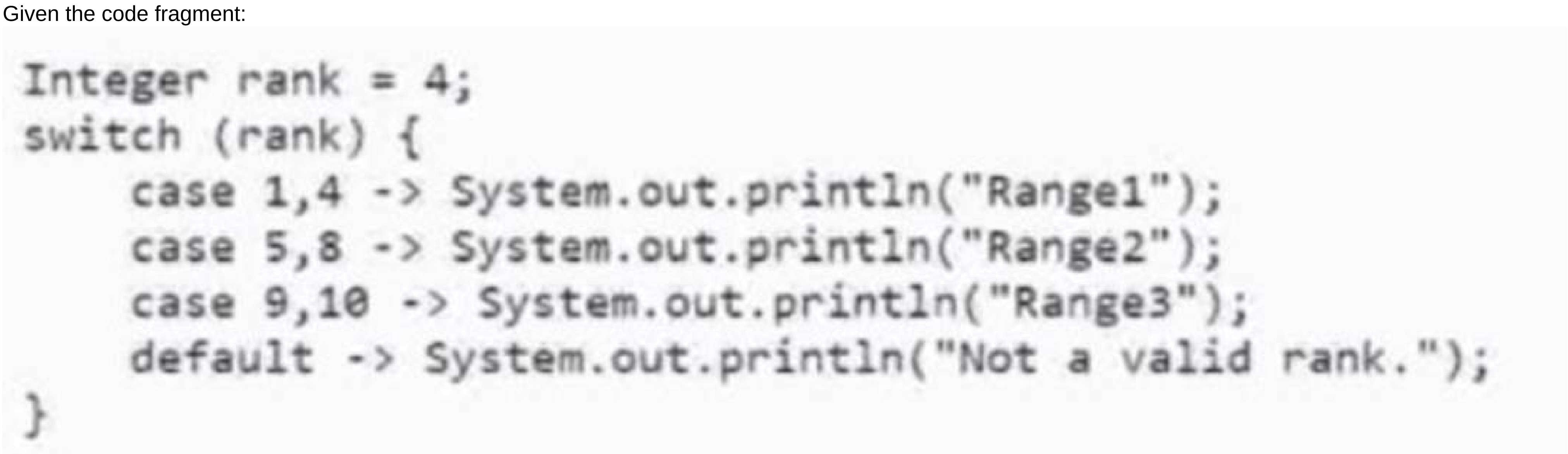
Which statement is true?

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 27
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?

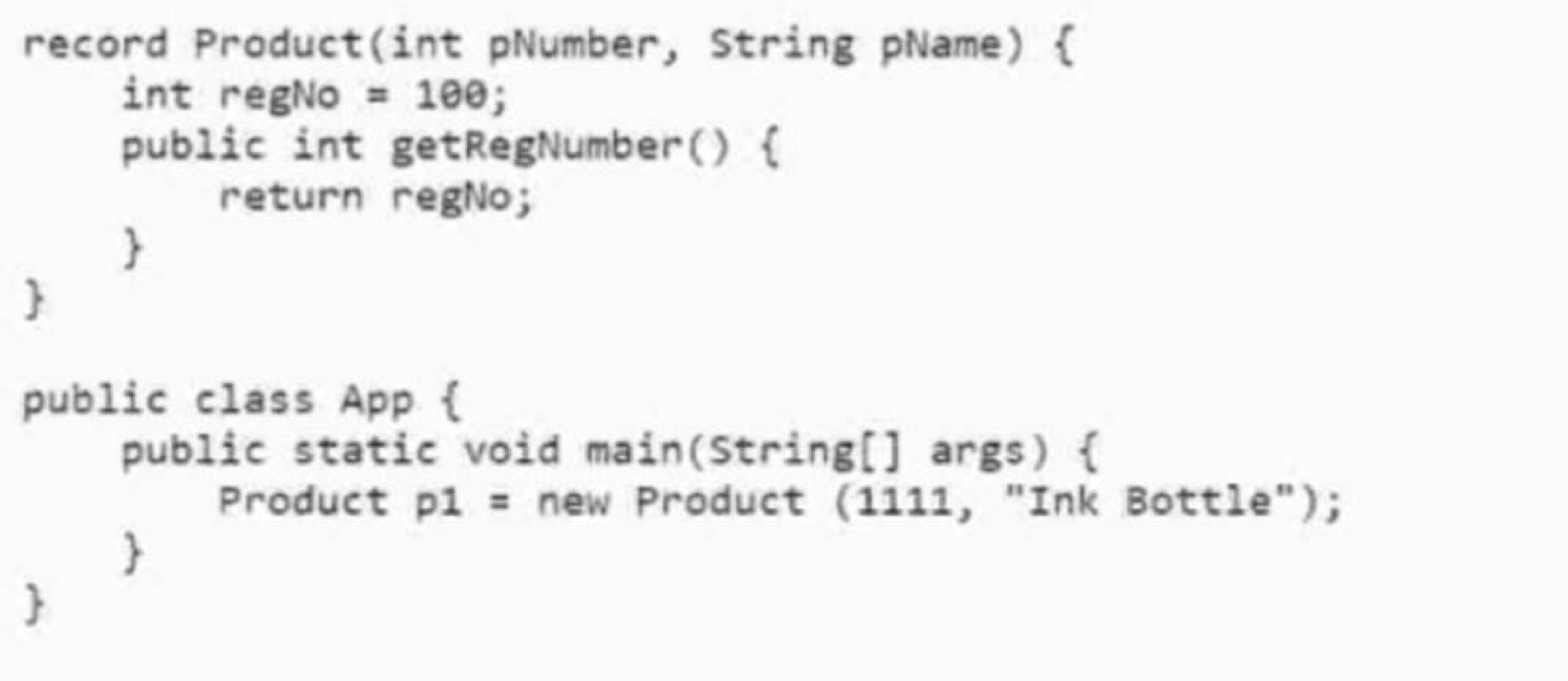
A. Range 1Range 2Range 3

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 31
Given the code fragment:



```
record Product(int pNumber, String pName) {
    int regNo = 100;
    public int getRegNumber() {
        return regNo;
    }
}

public class App {
    public static void main(String[] args) {
        Product p1 = new Product (1111, "Ink Bottle");
    }
}
```

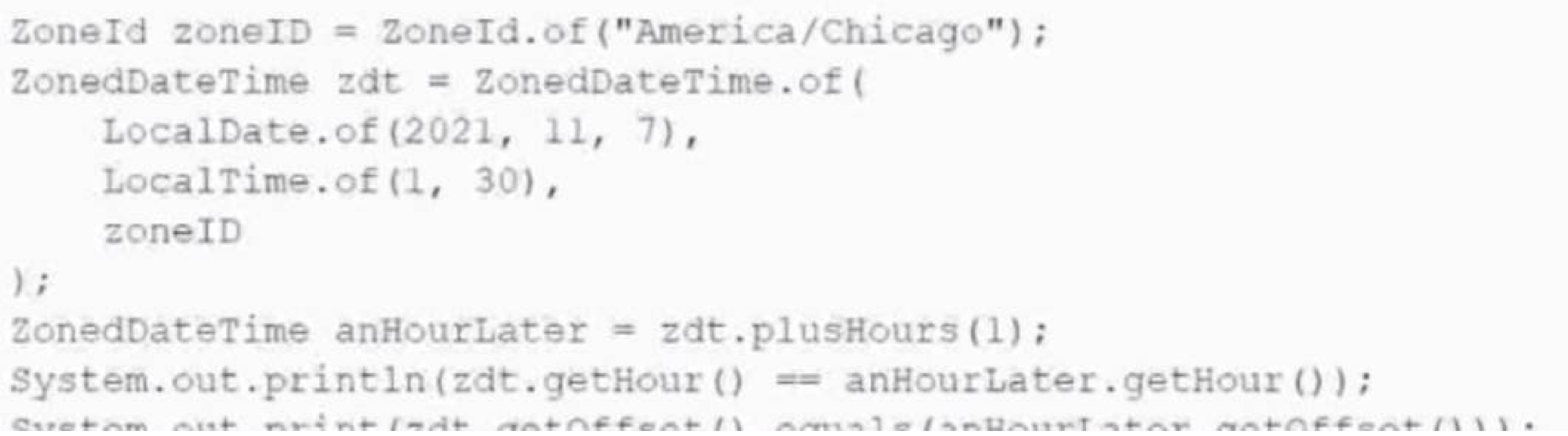
Which action enables the code to compile?

A. Replace record with void.
B. Remove the regNO initialization statement.
C. Make the regNo variable static.
D. Replace thye regNo variable static
E. Make the regNo variable public

Answer: E

Explanation:
The code will compile if the regNo variable is made public. This is because the regNo variable is being accessed in the main method of the App class, which is outside the scope of the Product class. Making the regNo variable public will allow it to be accessed from outside the class. 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/tutorial/java/javaOO/accesscontrol.html

NEW QUESTION 32
Daylight Saving Time (DST) is the practice of advancing clocks at the start of spring by one hour and adjusting them backward by one hour in autumn.
Considering that in 2021, DST in Chicago (Illinois) ended on November 7th at 2 AM, and given the fragment:



```
ZoneId zoneID = ZoneId.of("America/Chicago");
ZonedDateTime zdt = ZonedDateTime.of(
    LocalDate.of(2021, 11, 7),
    LocalTime.of(1, 30),
    zoneID
);
ZonedDateTime anHourLater = zdt.plusHours(1);
System.out.println(zdt.getHour() == anHourLater.getHour());
System.out.print(zdt.getOffset().equals(anHourLater.getOffset()));
```

What is the output?

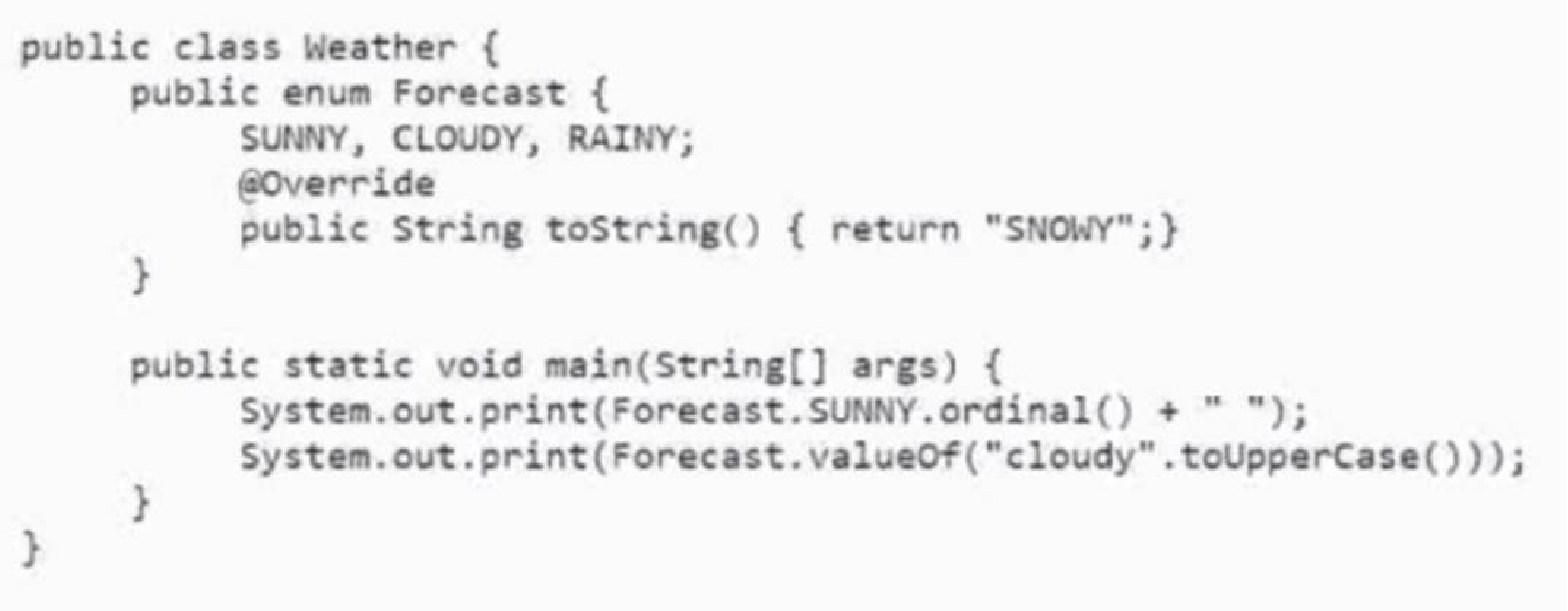
A. true false
B. False false
C. true true

D. false true

Answer: A

Explanation:
The answer is A because the code fragment uses the ZoneId and ZonedDateTime classes to create two date-time objects with the same local date-time but different zone offsets. The ZoneId class represents a time-zone ID, such as America/Chicago, and the ZonedDateTime class represents a date-time with a time-zone in the ISO-8601 calendar system. The code fragment creates two ZonedDateTime objects with the same local date-time of 2021-11-07T01:30, but different zone IDs of America/Chicago and UTC. The code fragment then compares the two objects using the equals and isEqual methods.
The equals method compares the state of two objects for equality. In this case, it compares the local date-time, zone offset, and zone ID of the two ZonedDateTime objects. Since the zone offsets and zone IDs are different, the equals method returns false.
The isEqual method compares the instant of two temporal objects for equality. In this case, it compares the instant of the two ZonedDateTime objects, which is derived from the local date-time and zone offset. Since DST in Chicago ended on November 7th at 2 AM in 2021, the local date-time of 2021-11-07T01:30 in America/Chicago corresponds to the same instant as 2021-11-07T06:30 in UTC. Therefore, the isEqual method returns true.
Hence, the output is true false. References:
? Oracle Certified Professional: Java SE 17 Developer
? Java SE 17 Developer
? OCP Oracle Certified Professional Java SE 17 Developer Study Guide
? ZoneId (Java Platform SE 8)
? ZonedDateTime (Java Platform SE 8)
? Time Zone & Clock Changes in Chicago, Illinois, USA
? Daylight Saving Time Changes 2023 in Chicago, USA

NEW QUESTION 35
Given:



```
public class Weather {
     public enum Forecast {
          SUNNY, CLOUDY, RAINY;
          @Override
          public String toString() { return "SNOWY";}
     }

     public static void main(String[] args) {
          System.out.print(Forecast.SUNNY.ordinal() + " ");
          System.out.print(Forecast.valueOf("cloudy".toUpperCase()));
     }
}
```

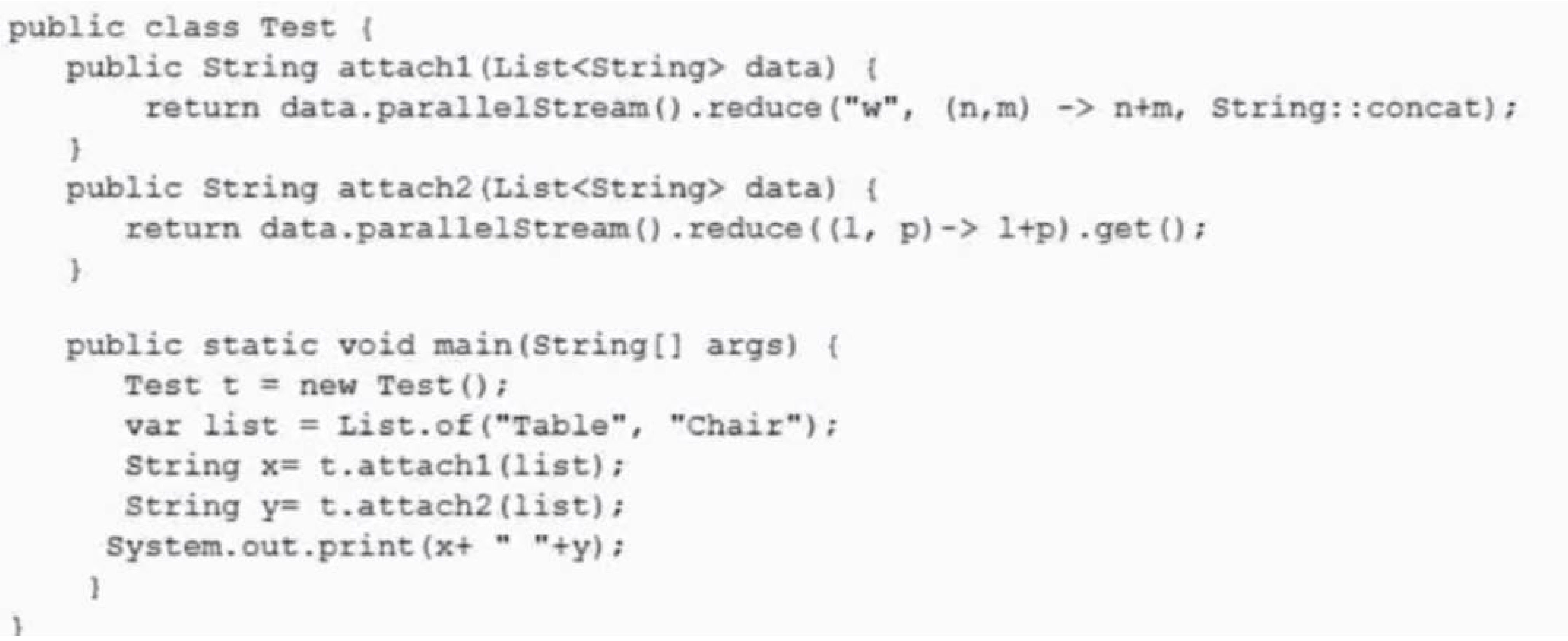
What is the result?

A. 1 RAINY
B. Compilation fails
C. 1 Snowy
D. 0 CLOUDY
E. 0 Snowy

Answer: E

Explanation:
The code is defining an enum class called Forecast with three values: SUNNY, CLOUDY, and RAINY. The toString() method is overridden to always return ??SNOWY??. In the main method, the ordinal value of SUNNY is printed, which is 0, followed by the value of CLOUDY converted to uppercase, which is ??CLOUDY??. However, since the
toString() method of Forecast returns ??SNOWY?? regardless of the actual value, the output
will be ??0 SNOWY??. References: Enum (Java SE 17 & JDK 17), Enum.EnumDesc (Java SE 17 & JDK 17)

NEW QUESTION 37
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);
    }
}
```

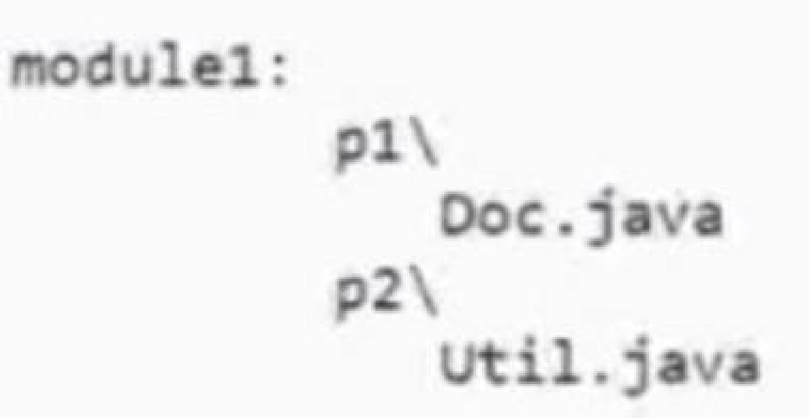
What is the result?

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

Answer: E

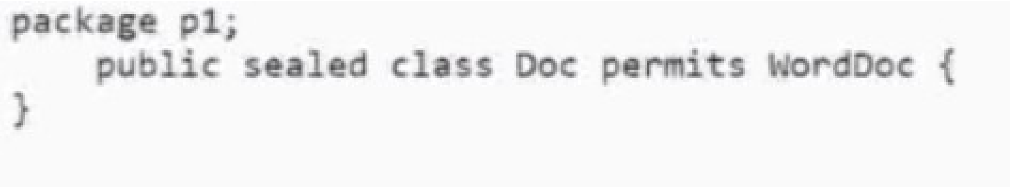
Explanation:
The code fragment will fail to compile because the class name and the constructor name do not match. The class name is Furniture, but the constructor name is Wtable. This will cause a syntax error. The correct way to define a constructor is to use the same name as the class name. Therefore,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

NEW QUESTION 40
Given the directory structure:



```
module1:
            p1\
                Doc.java
            p2\
                Util.java
```

Given the definition of the Doc class:



```
package p1;
    public sealed class Doc permits WordDoc {
}
```

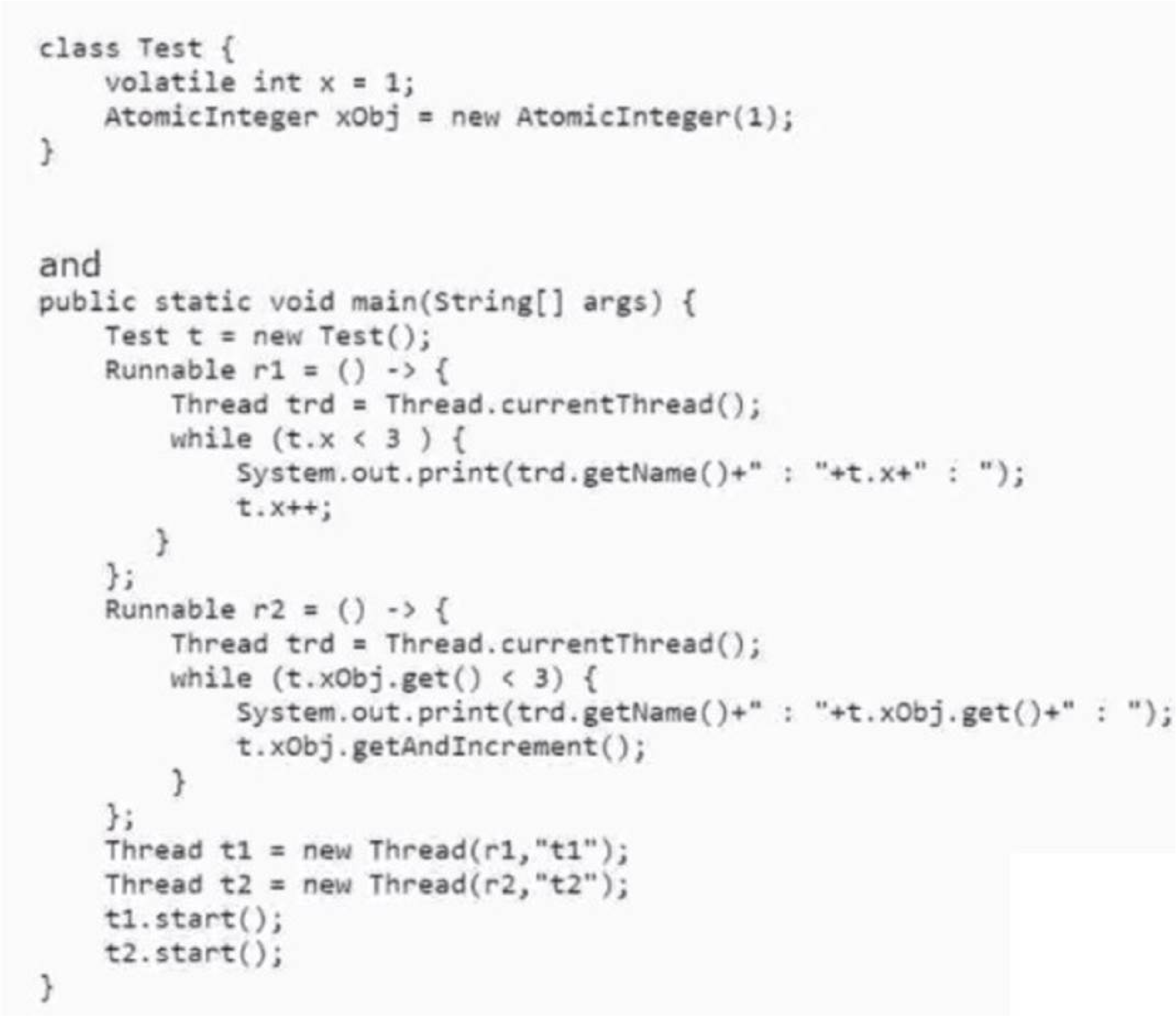
Which two are valid definition of the wordDoc class?

A. Package p1;Public non-sealed class wordDoc extends Doc ()
B. Package p1;Public class wordDoc extends Doc ()
C. Package p1, p2;Public non-sealed class WordDoc extends Doc ()
D. Package p1, p2;Public sealed class WordDoc extends Doc ()
E. Package p1,non-sealed abstract class WordDoc extends Doc ()
F. Package p1;Public final class WordDoc extends Doc ()

Answer: AF

Explanation:
The correct answer is A and F because the wordDoc class must be a non-sealed class or a final class to extend the sealed Doc class. Option B is incorrect because the wordDoc class must be non-sealed or final. Option C is incorrect because the wordDoc class cannot be in a different package than the Doc class. Option D is incorrect because the wordDoc class cannot be a sealed class. Option E is incorrect because the wordDoc class cannot be an abstract class.
References: Oracle Certified Professional: Java SE 17 Developer, 3 Sealed Classes - Oracle Help Center

NEW QUESTION 41
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 45
......