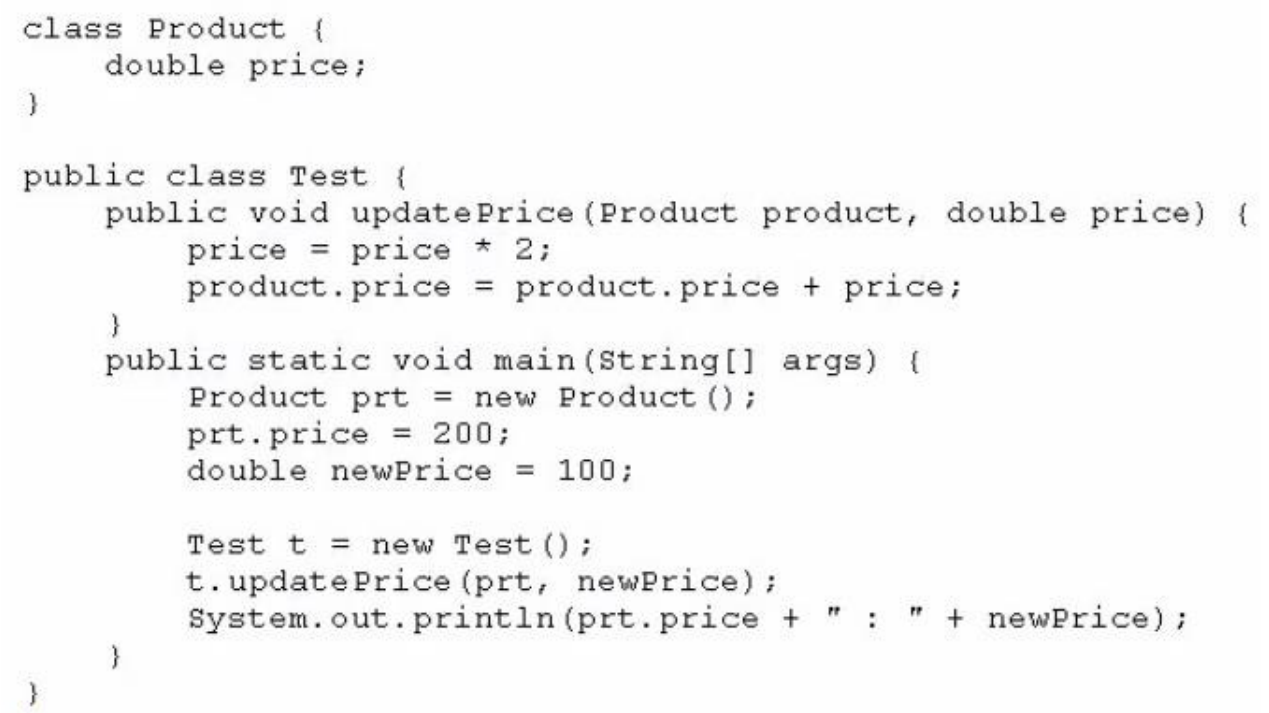
Examen
Desarrollador Java

1. What is Java?
2. What are the concepts of OOP?
3. Mention some features of Java
4. What is Abstraction?
5. What is Encapsulation?
6. What are the differences between Abstraction and Encapsulation?
7. What is Polymorphism?
8. What are the types of Polymorphism?
9. What does the "static" keyword mean?
10. Can you override private or static method in Java?
11. Can you access non static variable in static context?
12. What is Autoboxing and Unboxing?
13. What is a Constructor?
14. What is Constructor Overloading?
15. What is the difference between an Interface and an Abstract class?
16. What are pass by reference and pass by value?
17. What is Local Variable and Instance Variable?
18. What are the different access modifiers available in Java?
19. Explain different ways of creating a thread. Which one would you prefer and why?
20. Explain the available thread states in a high-level.
21. What is the difference between a method and block that are synchronized?
22. What is the importance of hashCode() and equals() methods?
23. What is the purpose of garbage collection in Java, and when is it used?
24. When is the finalize() called? What is the purpose of finalization?
25. What are the differences between Checked Exception and Unchecked Exception?
26. What is the difference between Exception and Error in java?
27. What is the importance of finally block in exception handling?
28. What purpose does the keywords final, finally, and finalize fulfill?
29. What are the JDBC API components?
30. What is the advantage of PreparedStatement over Statement?
31. What is the use of CallableStatement?
32. What is a Servlet?
33. Explain the architecture of a Servlet.
34. What is the difference between doGet() and doPost()?
35. What is the difference between session and cookie?
36. What’s the difference between sendRedirect and forward methods?
37. What is Request Dispatcher?
38. What is a JSP Page?
39. What are the advantages of JSP?
40. What are JSP actions?
41. Qué son los scriptlets?
42. What are the implicit objects within a JSP?

Given:

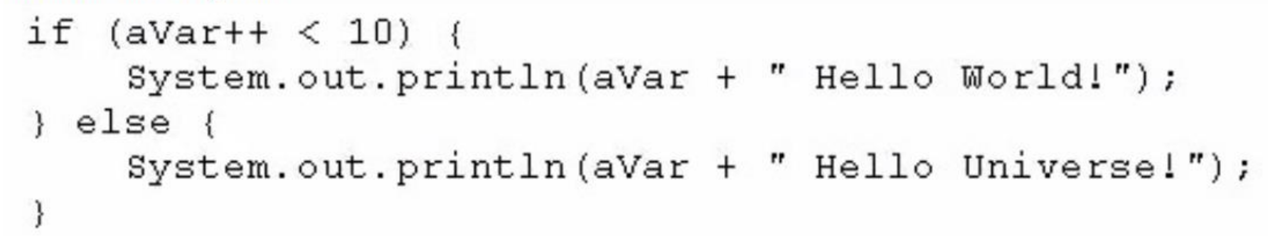


```
class Product {
    double price;
}

public class Test {
    public void updatePrice(Product product, double price) {
        price = price * 2;
        product.price = product.price + price;
    }
    public static void main(String[] args) {
        Product prt = new Product();
        prt.price = 200;
        double newPrice = 100;

        Test t = new Test();
        t.updatePrice(prt, newPrice);
        System.out.println(prt.price + " : " + newPrice);
    }
}
```

What is the result?

A. 200.0 : 100.0
B. 400.0 : 200.0
C. 400.0 : 100.0
D. Compilation fails.

Given the code fragment:

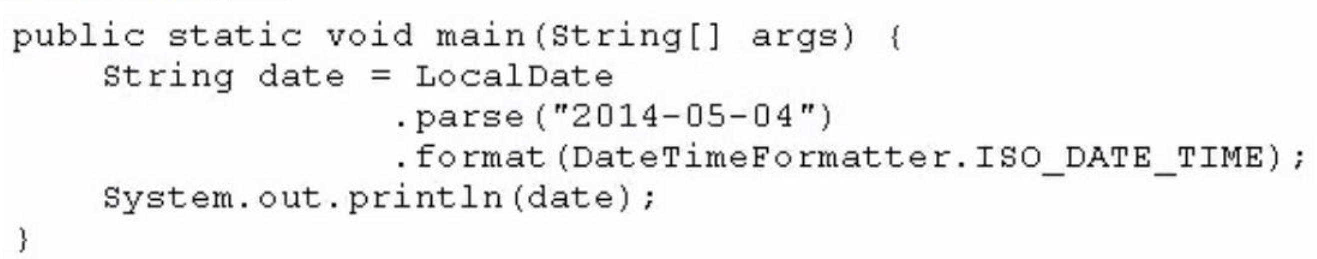


```
if (aVar++ < 10) {
    System.out.println(aVar + " Hello World!");
} else {
    System.out.println(aVar + " Hello Universe!");
}
```

What is the result if the integer aVar is 9?

A. 10 Hello World!
B. Hello Universe!
C. Hello World!
D. Compilation fails.

Given the code fragment:

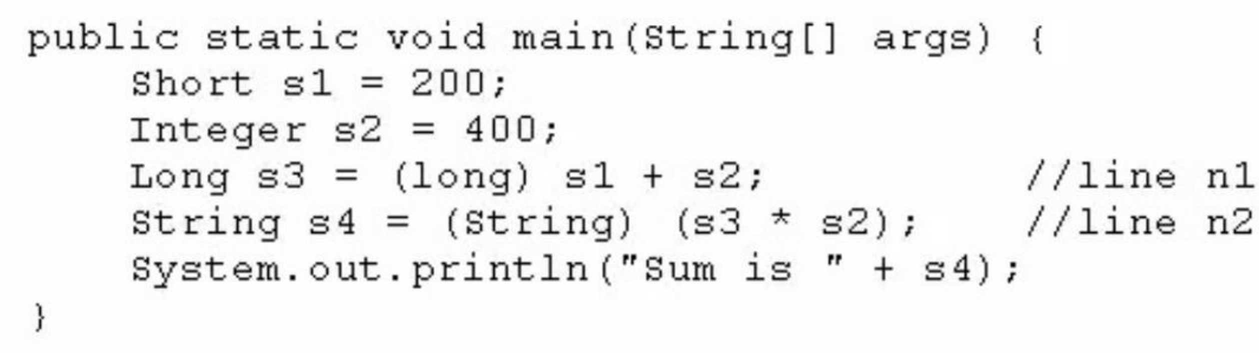


```
public static void main(String[] args) {
    String date = LocalDate
                  .parse("2014-05-04")
                  .format(DateTimeFormatter.ISO_DATE_TIME);
    System.out.println(date);
}
```

What is the result?

A. May 04, 2014T00:00:00.000
B. 2014-05-04T00:00: 00. 000
C. 5/4/14T00:00:00.000
D. An exception is thrown at runtime.

Given the code fragment:

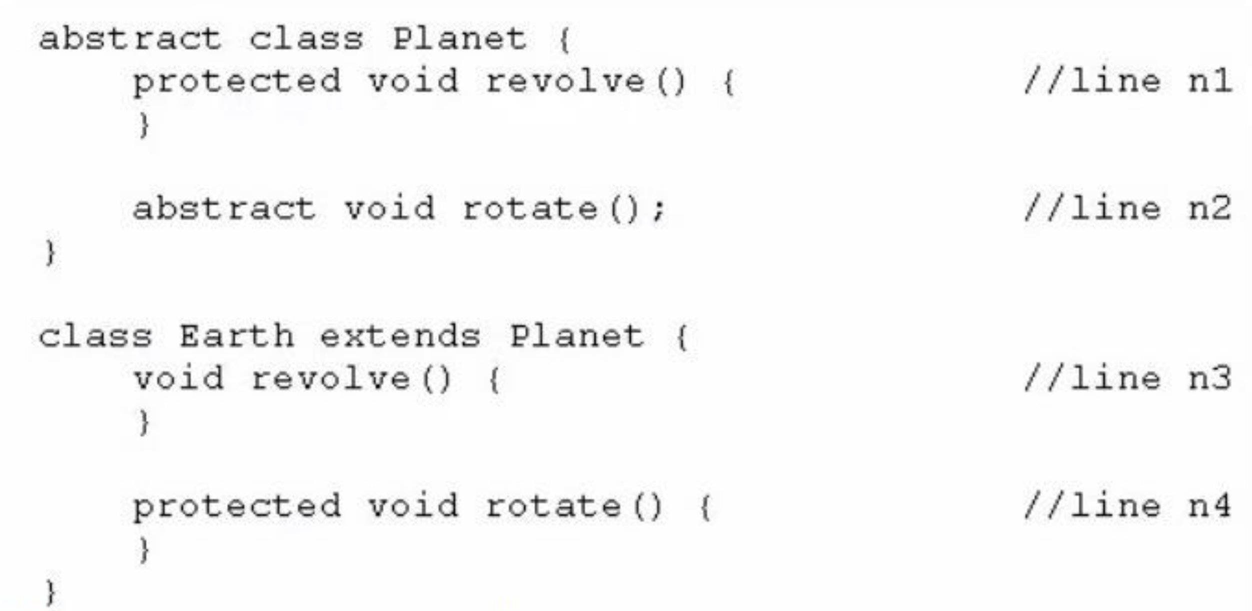


```
public static void main(String[] args) {
    Short s1 = 200;
    Integer s2 = 400;
    Long s3 = (long) s1 + s2;            //line n1
    String s4 = (String) (s3 * s2);      //line n2
    System.out.println("Sum is " + s4);
}
```

What is the result?

A. Sum is 600
B. Compilation fails at line n1.
C. Compilation fails at line n2.
D. A ClassCastException is thrown at line n1.
E. A ClassCastException is thrown at line n2.

Given the code fragment:

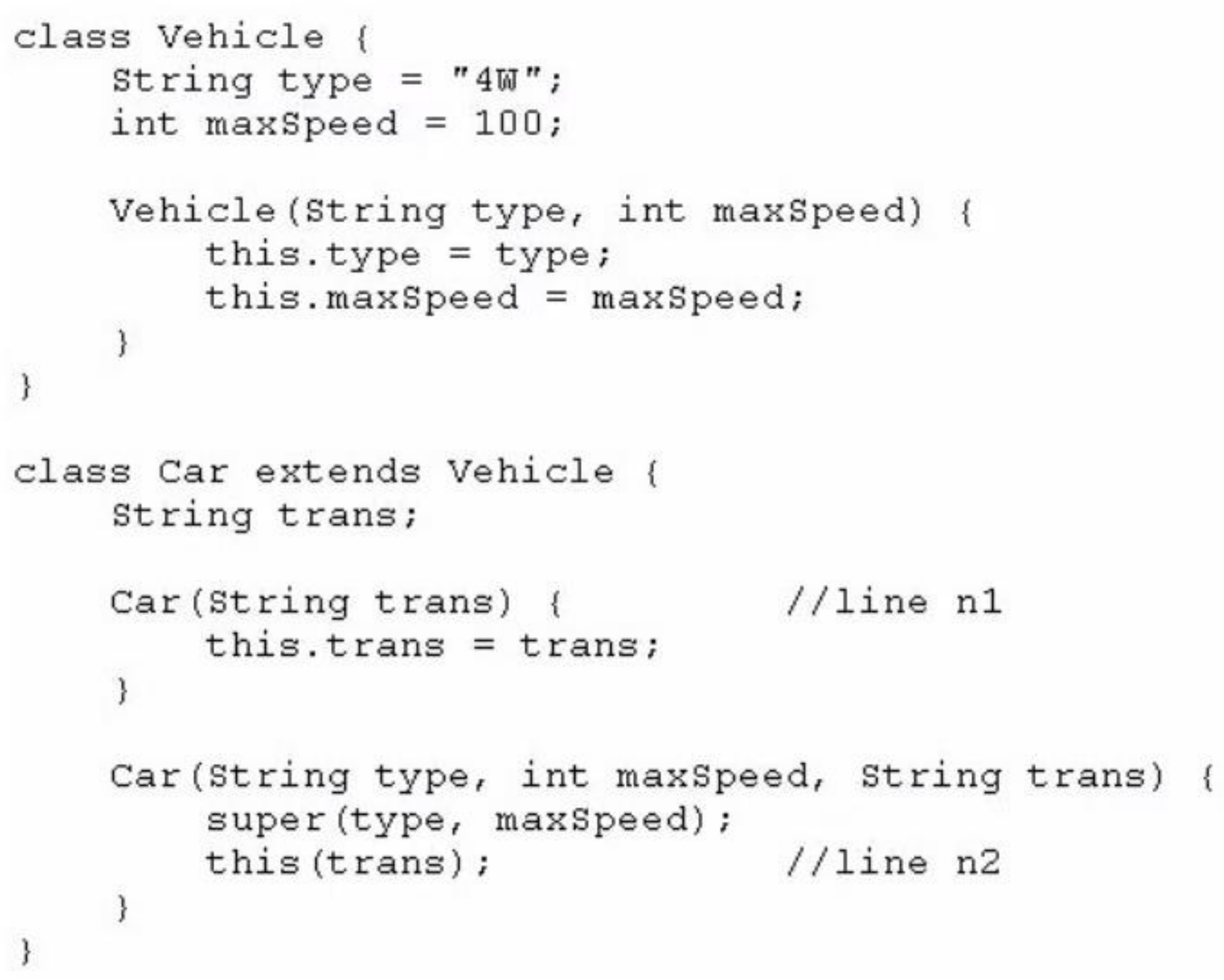


```
abstract class Planet {
    protected void revolve() {                    //line n1
    }

    abstract void rotate();                       //line n2
}

class Earth extends Planet {
    void revolve() {                              //line n3
    }

    protected void rotate() {                     //line n4
    }
}
```

Which two modifications, made independently, enable the code to compile?

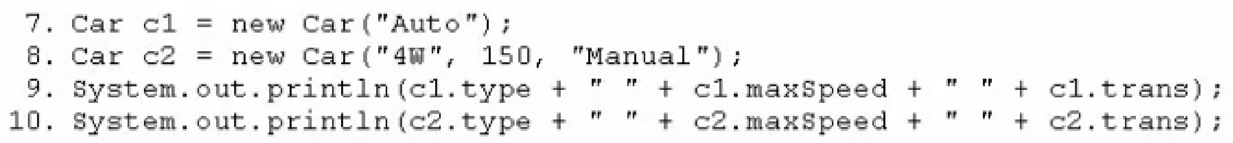
A. Make the method at line n1 public.
B. Make the method at line n2 public.
C. Make the method at line n3 public.
D. Make the method at line n3 protected.
E. Make the method at line n4 public.

Given the code fragment:



```
class Vehicle {
    String type = "4W";
    int maxSpeed = 100;

    Vehicle(String type, int maxSpeed) {
        this.type = type;
        this.maxSpeed = maxSpeed;
    }
}

class Car extends Vehicle {
    String trans;

    Car(String trans) {         //line n1
        this.trans = trans;
    }

    Car(String type, int maxSpeed, String trans) {
        super(type, maxSpeed);
        this(trans);            //line n2
    }
}
```

And given the code fragment:

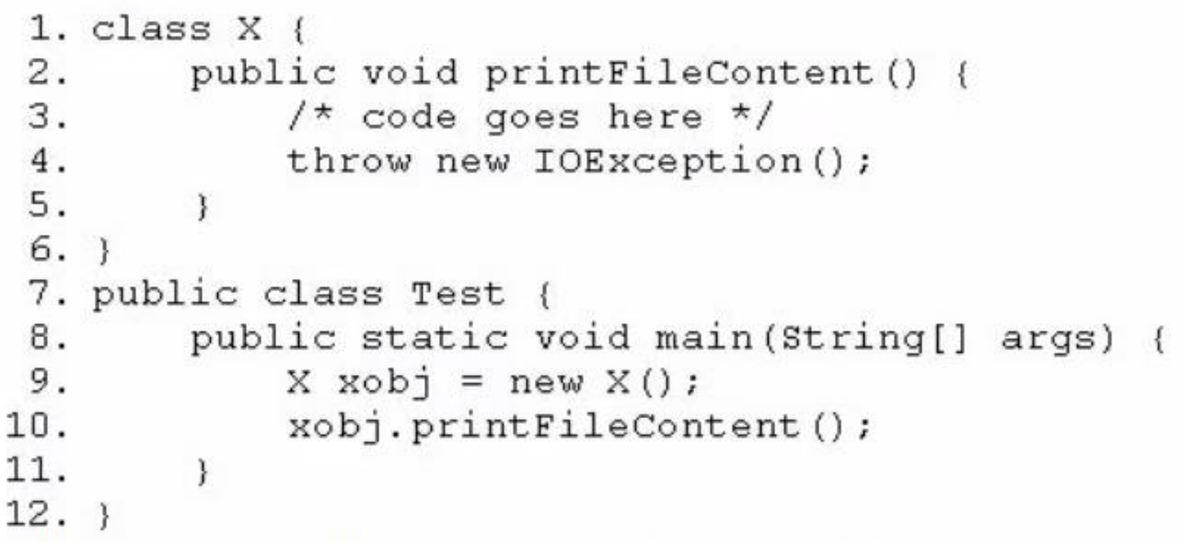


```
Car c1 = new Car("Auto");
Car c2 = new Car("4W", 150, "Manual");
System.out.println(c1.type + " " + c1.maxSpeed + " " + c1.trans);
System.out.println(c2.type + " " + c2.maxSpeed + " " + c2.trans);
```

What is the result?

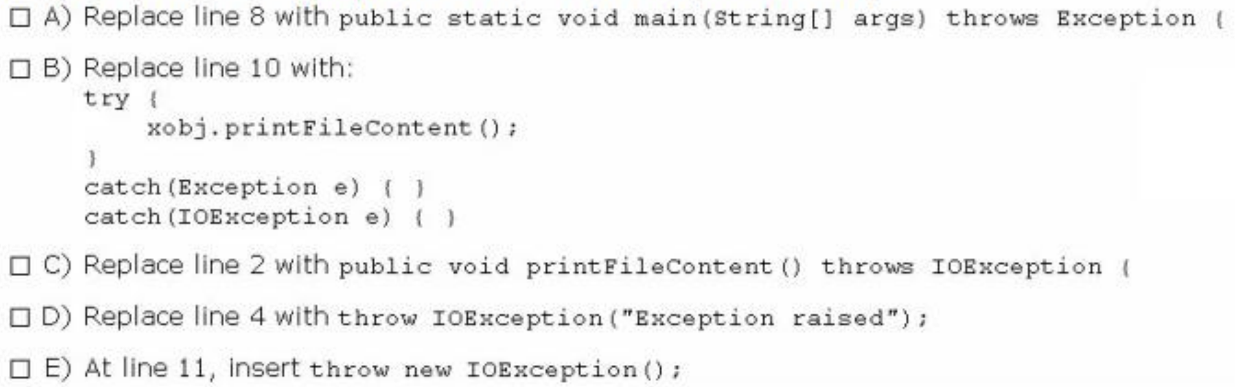
A. 4W 100 Auto4W 150 Manual
B. Null 0 Auto4W 150 Manual
C. Compilation fails only at line n1
D. Compilation fails only at line n2
E. Compilation fails at both line n1 and line n2

Given the code fragment:



```
 1. class X {
 2.     public void printFileContent() {
 3.         /* code goes here */
 4.         throw new IOException();
 5.     }
 6. }
 7. public class Test {
 8.     public static void main(String[] args) {
 9.         X xobj = new X();
10.         xobj.printFileContent();
11.     }
12. }
```

Which two modifications should you make so that the code compiles successfully?



□ A) Replace line 8 with `public static void main(String[] args) throws Exception {`

□ B) Replace line 10 with:

```
try {
    xobj.printFileContent();
}
catch(Exception e) { }
catch(IOException e) { }
```

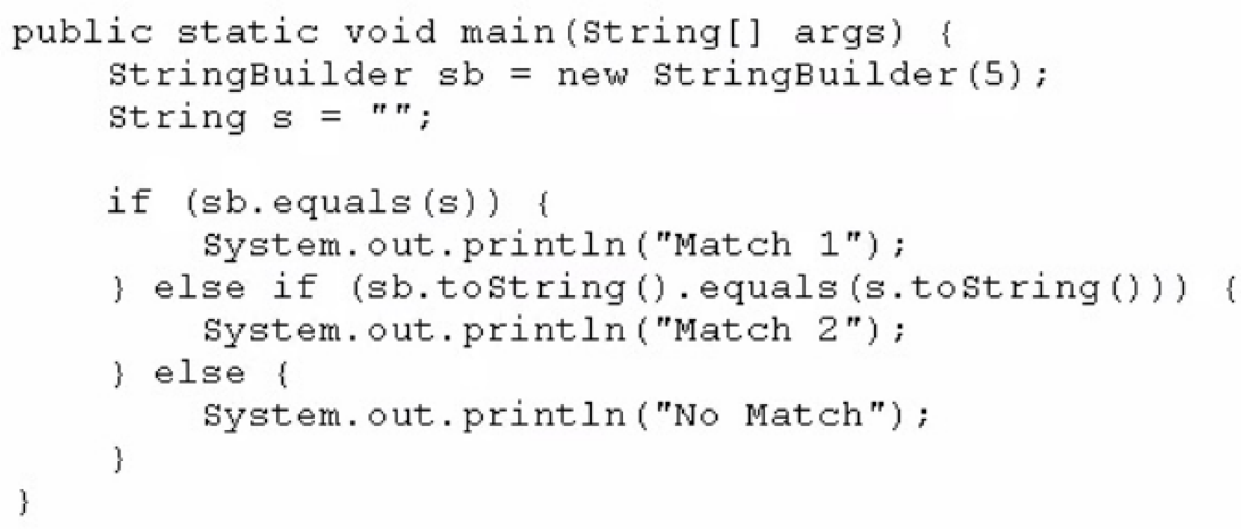
□ C) Replace line 2 with `public void printFileContent() throws IOException {`

□ D) Replace line 4 with `throw IOException("Exception raised");`

□ E) At line 11, insert `throw new IOException();`

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

Given the code fragment:

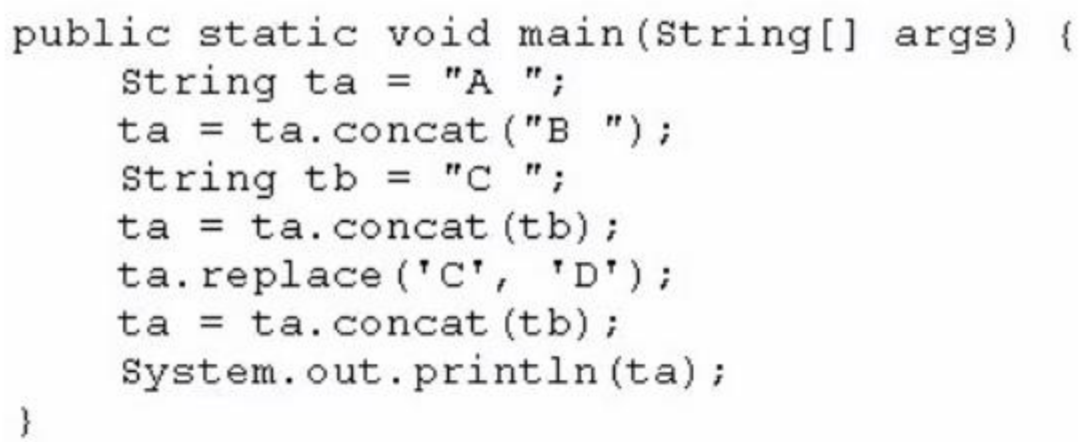


```
public static void main(String[] args) {
    StringBuilder sb = new StringBuilder(5);
    String s = "";

    if (sb.equals(s)) {
        System.out.println("Match 1");
    } else if (sb.toString().equals(s.toString())) {
        System.out.println("Match 2");
    } else {
        System.out.println("No Match");
    }
}
```

What is the result?

A. Match 1
B. Match 2
C. No Match
D. A NullPointerException is thrown at runtime.

Given the code fragment:



```
public static void main(String[] args) {
    String ta = "A ";
    ta = ta.concat("B ");
    String tb = "C ";
    ta = ta.concat(tb);
    ta.replace('C', 'D');
    ta = ta.concat(tb);
    System.out.println(ta);
}
```

What is the result?

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