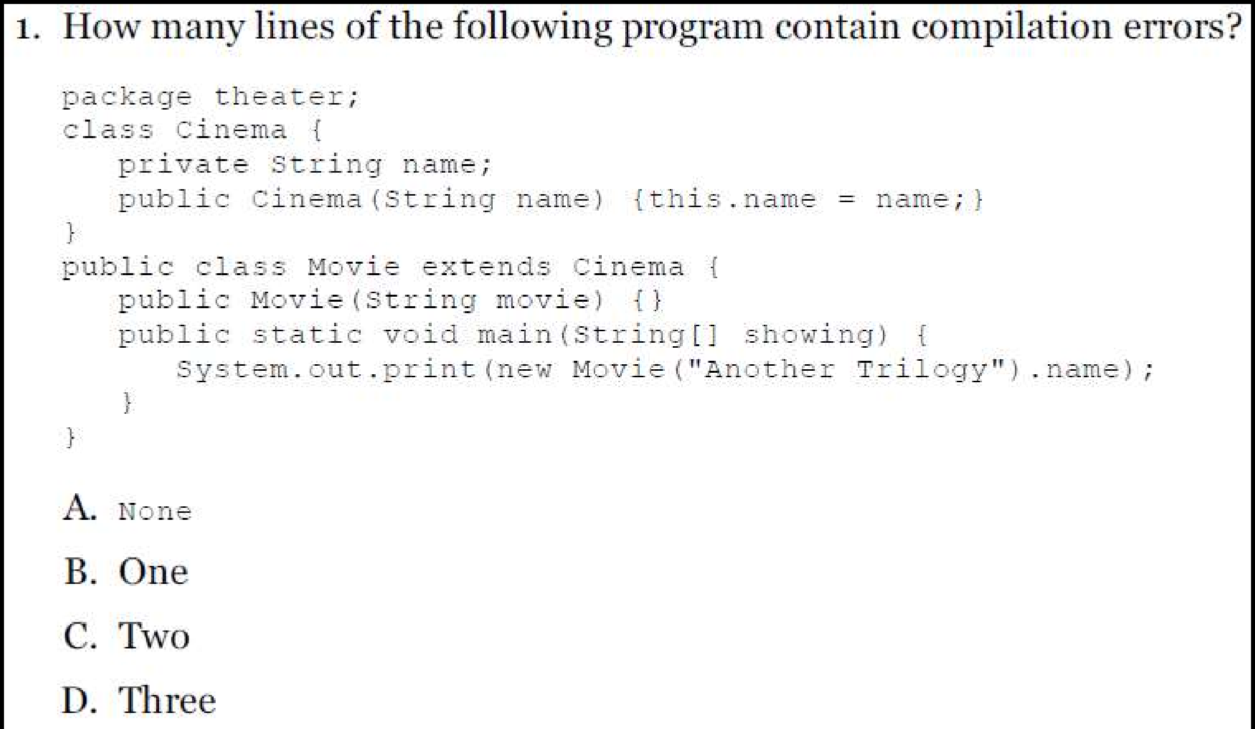
Deloitte.

Exercicios 2

Janeiro 2022

MAKING AN IMPACT THAT MATTERS since 1845

Deloitte.



1. How many lines of the following program contain compilation errors?

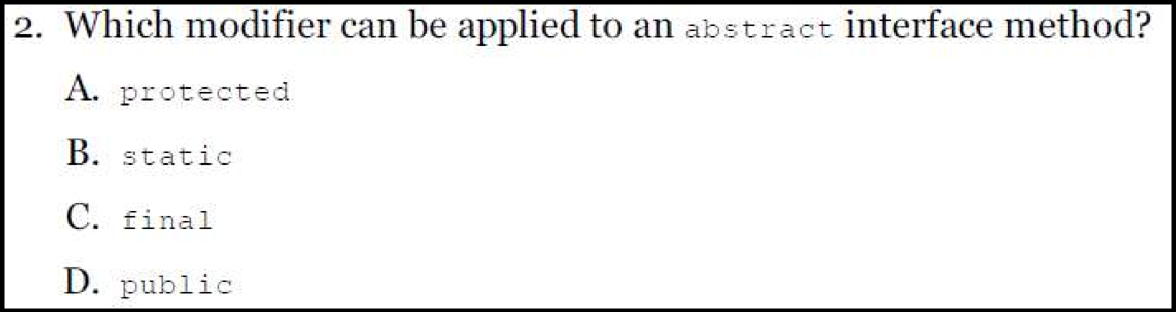
```
package theater;
class Cinema {
   private String name;
   public Cinema(String name) {this.name = name;}
}
public class Movie extends Cinema {
   public Movie(String movie) {}
   public static void main(String[] showing) {
      System.out.print(new Movie("Another Trilogy").name);
   }
}
```

A. None

B. One

C. Two

D. Three



2. Which modifier can be applied to an `abstract` interface method?

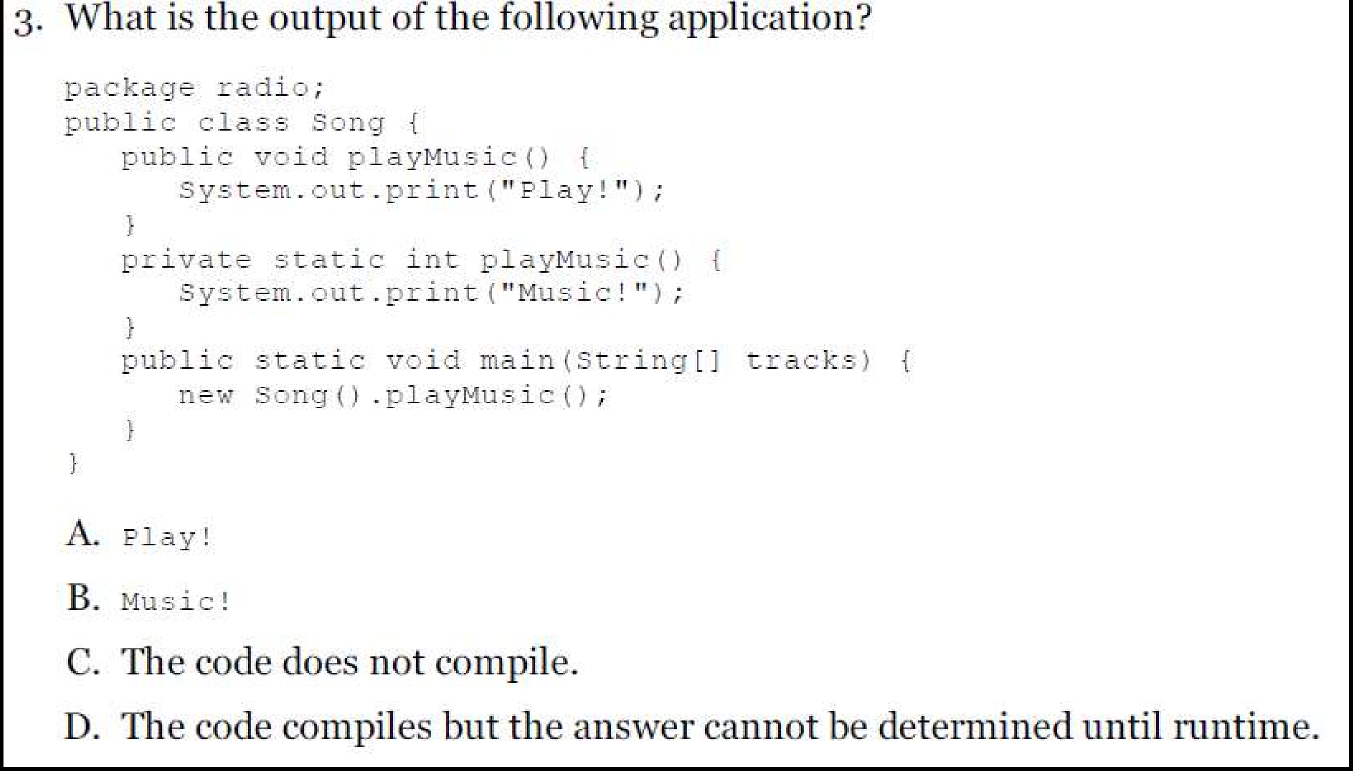
   A. `protected`

   B. `static`

   C. `final`

   D. `public`

Deloitte.



3. What is the output of the following application?

```
package radio;
public class Song {
   public void playMusic() {
      System.out.print("Play!");
   }
   private static int playMusic() {
      System.out.print("Music!");
   }
   public static void main(String[] tracks) {
      new Song().playMusic();
   }
}
```

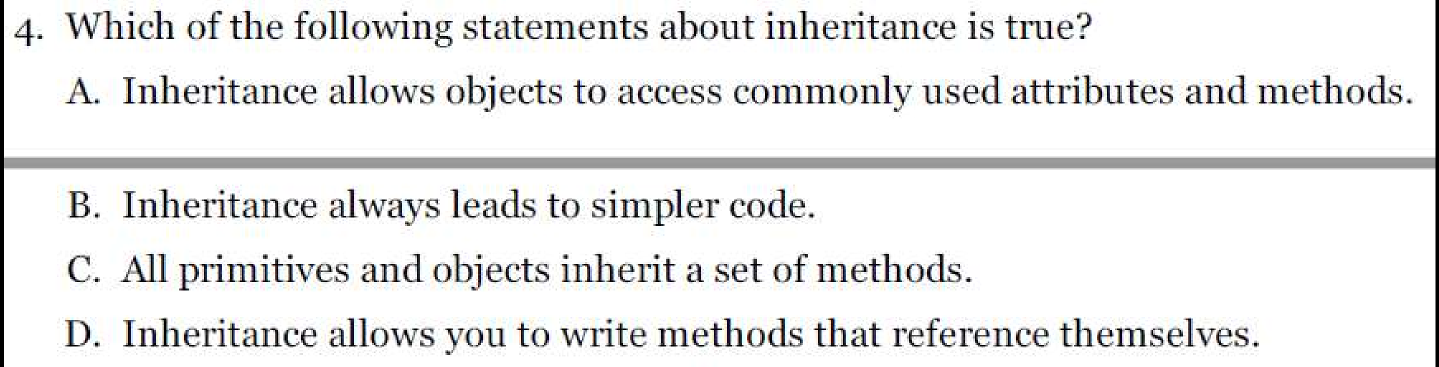
A. `Play!`

B. `Music!`

C. The code does not compile.

D. The code compiles but the answer cannot be determined until runtime.

Deloitte.



4. Which of the following statements about inheritance is true?

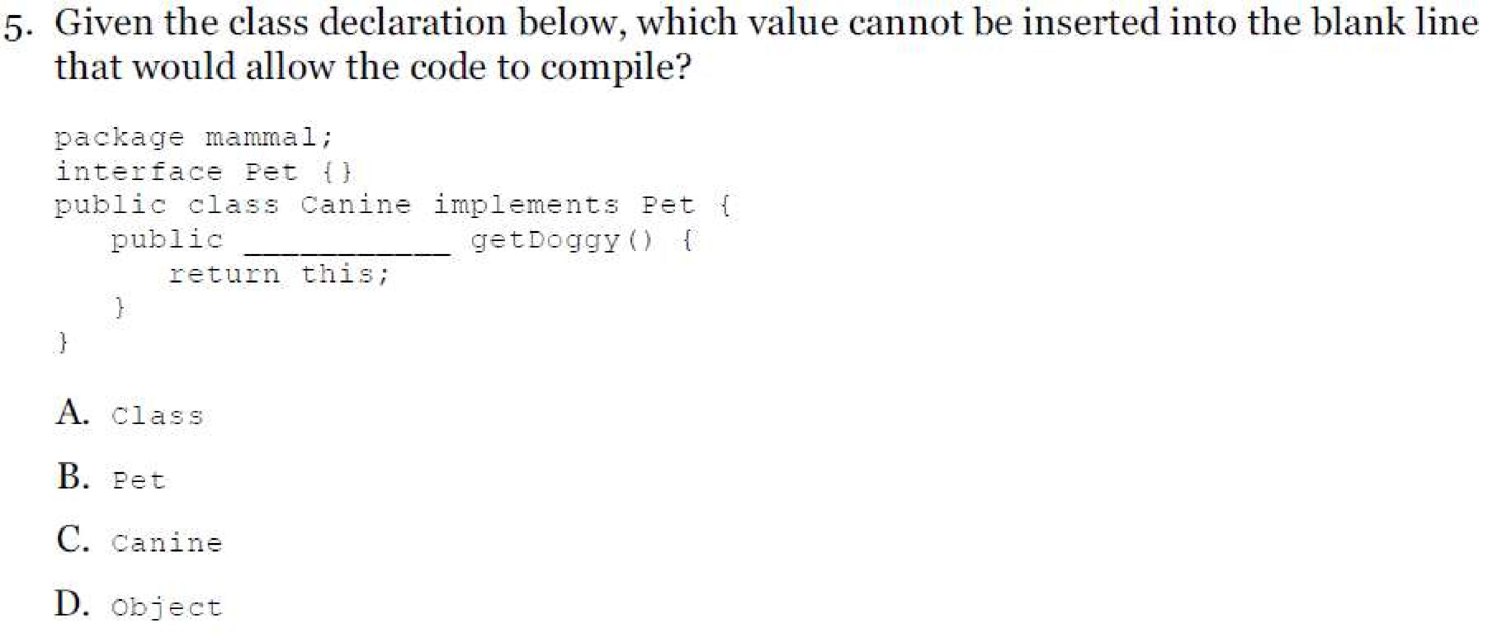
   A. Inheritance allows objects to access commonly used attributes and methods.

   B. Inheritance always leads to simpler code.

   C. All primitives and objects inherit a set of methods.

   D. Inheritance allows you to write methods that reference themselves.

Deloitte.



5. Given the class declaration below, which value cannot be inserted into the blank line that would allow the code to compile?

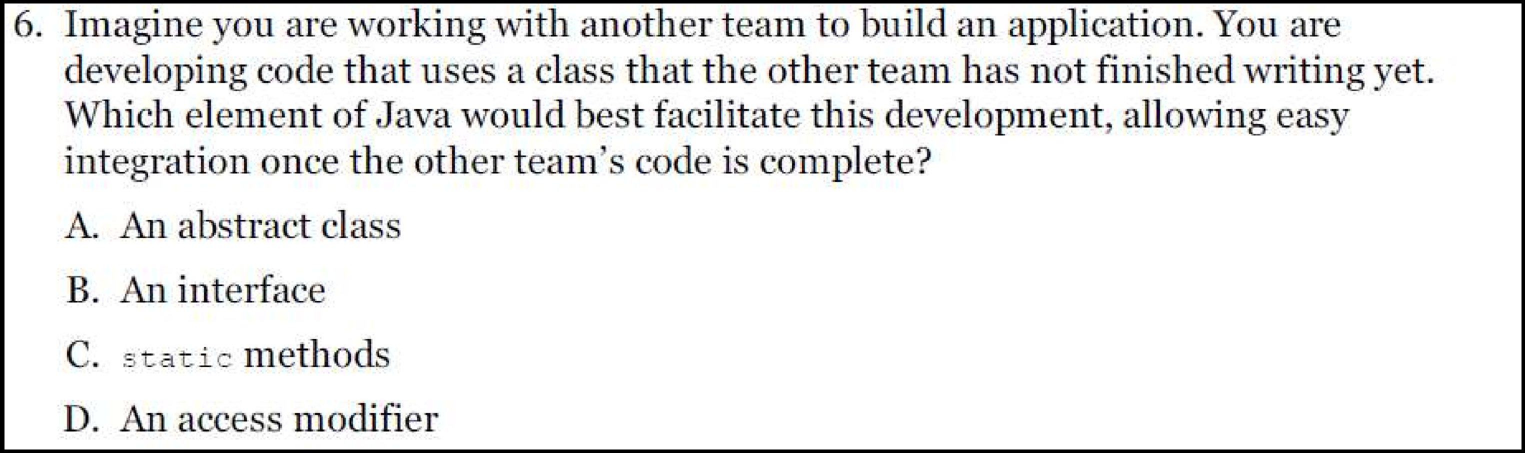
```
package mammal;
interface Pet {}
public class Canine implements Pet {
   public ___________ getDoggy() {
      return this;
   }
}
```

A. `Class`

B. `Pet`

C. `Canine`

D. `Object`



6. Imagine you are working with another team to build an application. You are developing code that uses a class that the other team has not finished writing yet. Which element of Java would best facilitate this development, allowing easy integration once the other team's code is complete?

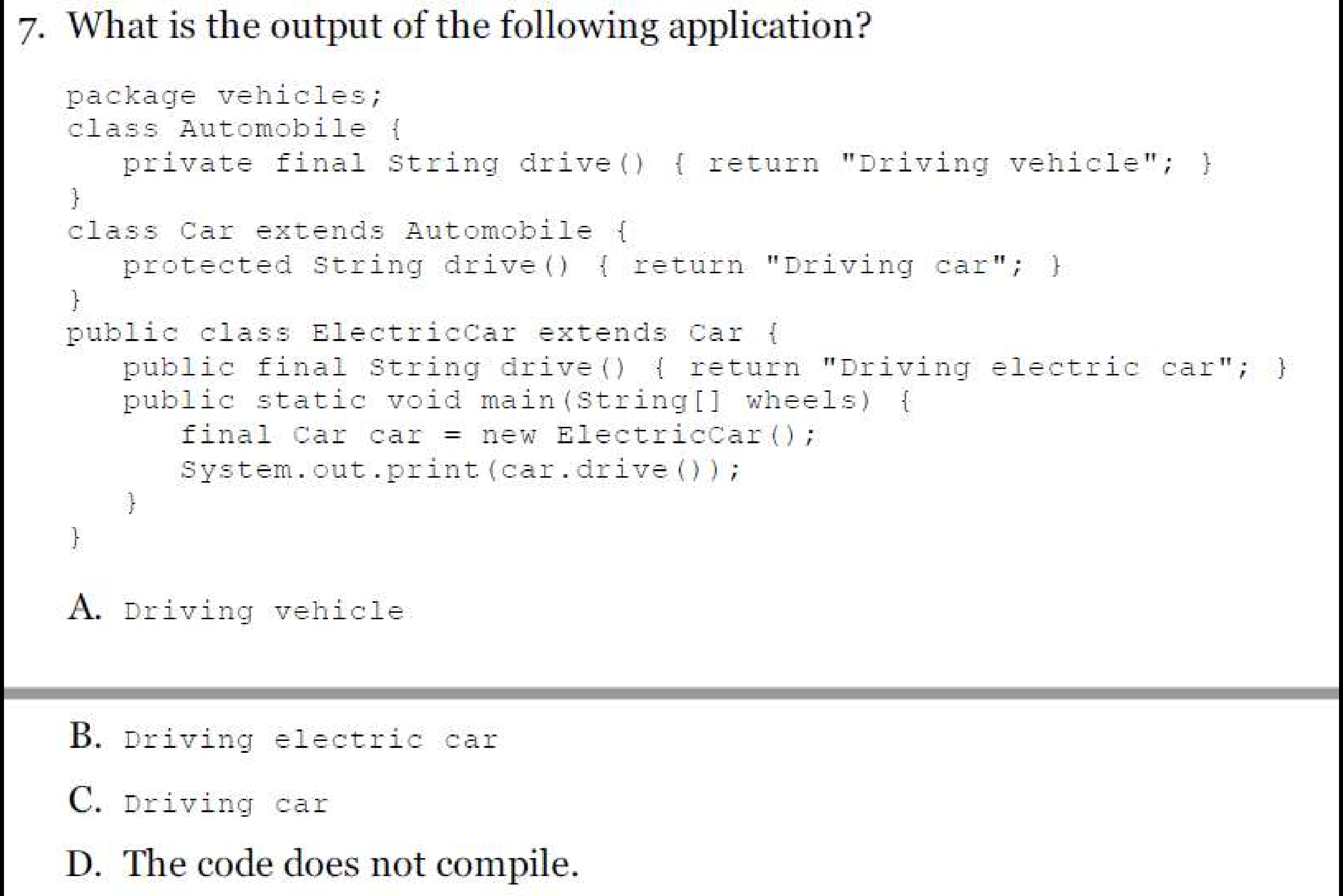
   A. An abstract class

   B. An interface

   C. `static` methods

   D. An access modifier

Deloitte.



7. What is the output of the following application?

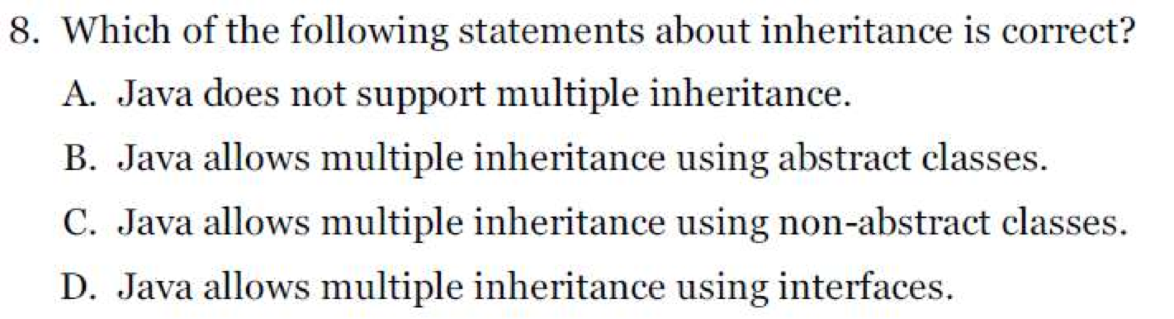
```
package vehicles;
class Automobile {
   private final String drive() { return "Driving vehicle"; }
}
class Car extends Automobile {
   protected String drive() { return "Driving car"; }
}
public class ElectricCar extends Car {
   public final String drive() { return "Driving electric car"; }
   public static void main(String[] wheels) {
      final Car car = new ElectricCar();
      System.out.print(car.drive());
   }
}
```

A. `Driving vehicle`

B. `Driving electric car`

C. `Driving car`

D. The code does not compile.



8. Which of the following statements about inheritance is correct?

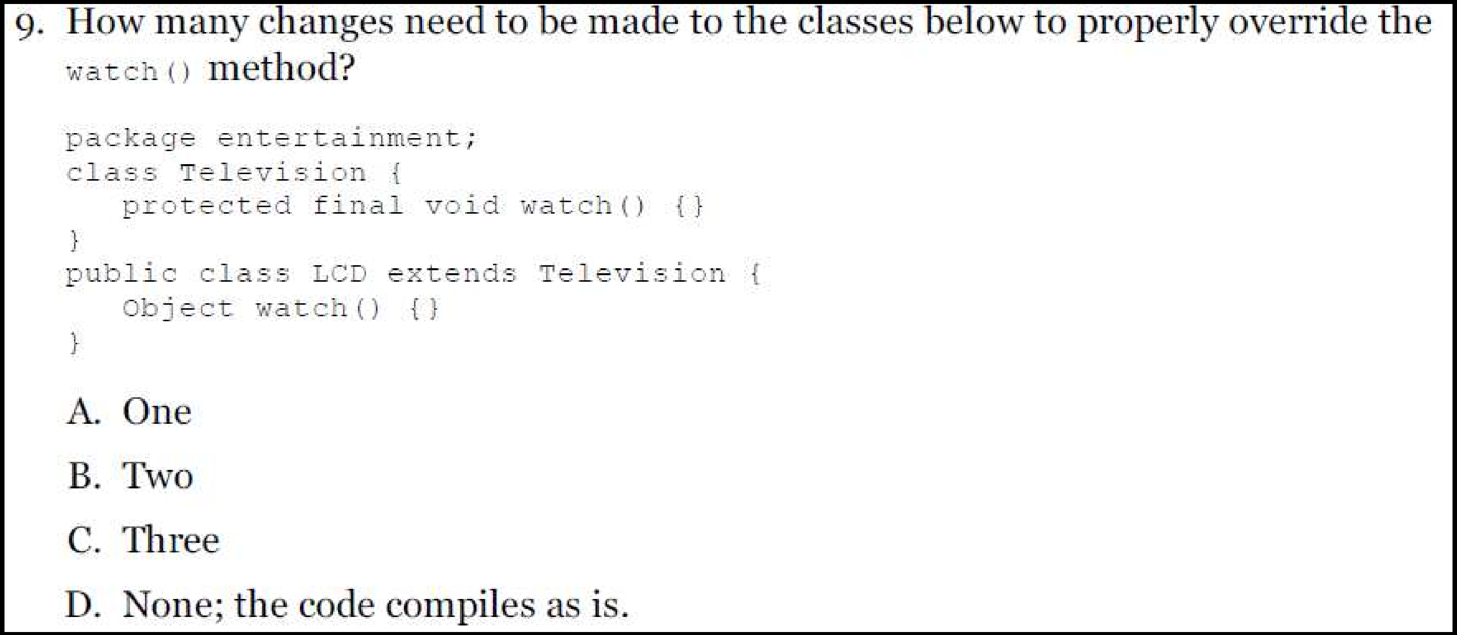
   A. Java does not support multiple inheritance.

   B. Java allows multiple inheritance using abstract classes.

   C. Java allows multiple inheritance using non-abstract classes.

   D. Java allows multiple inheritance using interfaces.

Deloitte.



9. How many changes need to be made to the classes below to properly override the `watch()` method?

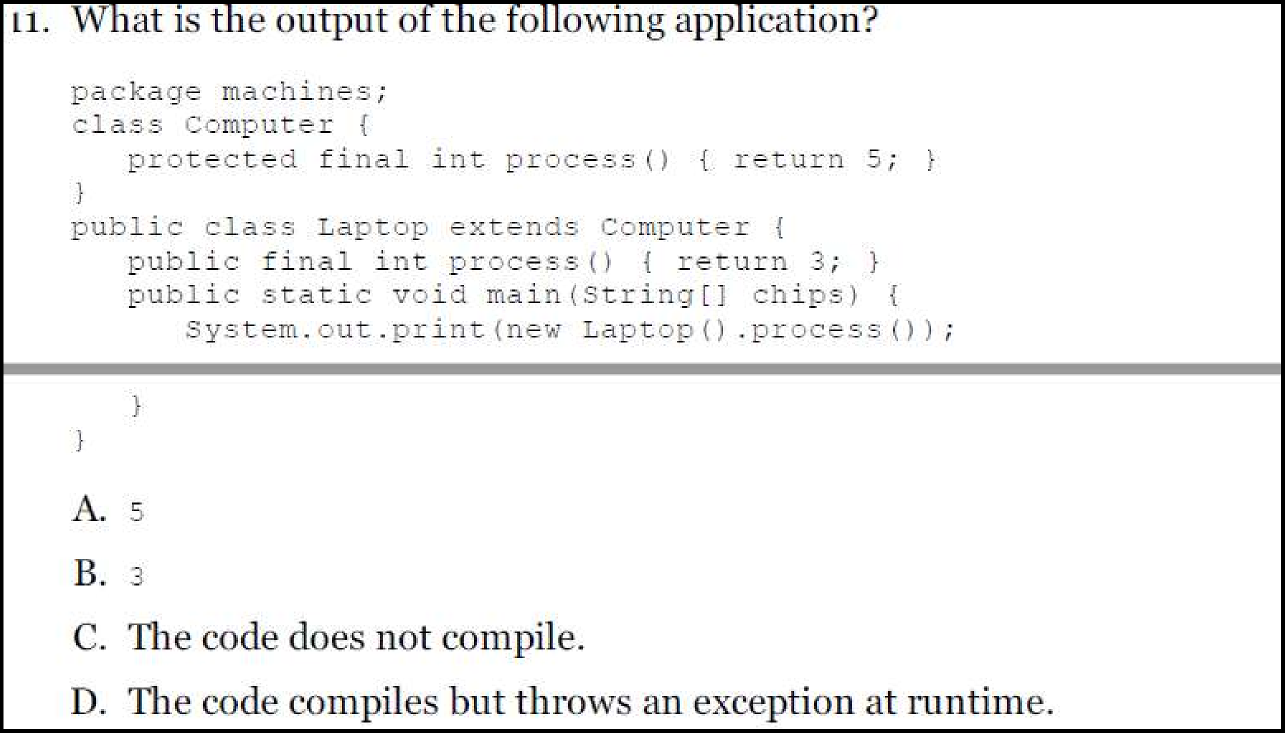
```
package entertainment;
class Television {
   protected final void watch() {}
}
public class LCD extends Television {
   Object watch() {}
}
```

A. One

B. Two

C. Three

D. None; the code compiles as is.



11. What is the output of the following application?

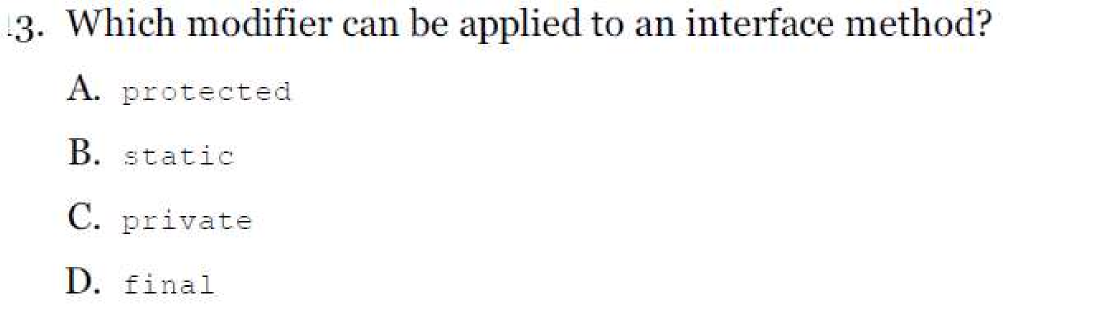
```
package machines;
class Computer {
   protected final int process() { return 5; }
}
public class Laptop extends Computer {
   public final int process() { return 3; }
   public static void main(String[] chips) {
      System.out.print(new Laptop().process());
   }
}
```

A. 5

B. 3

C. The code does not compile.

D. The code compiles but throws an exception at runtime.



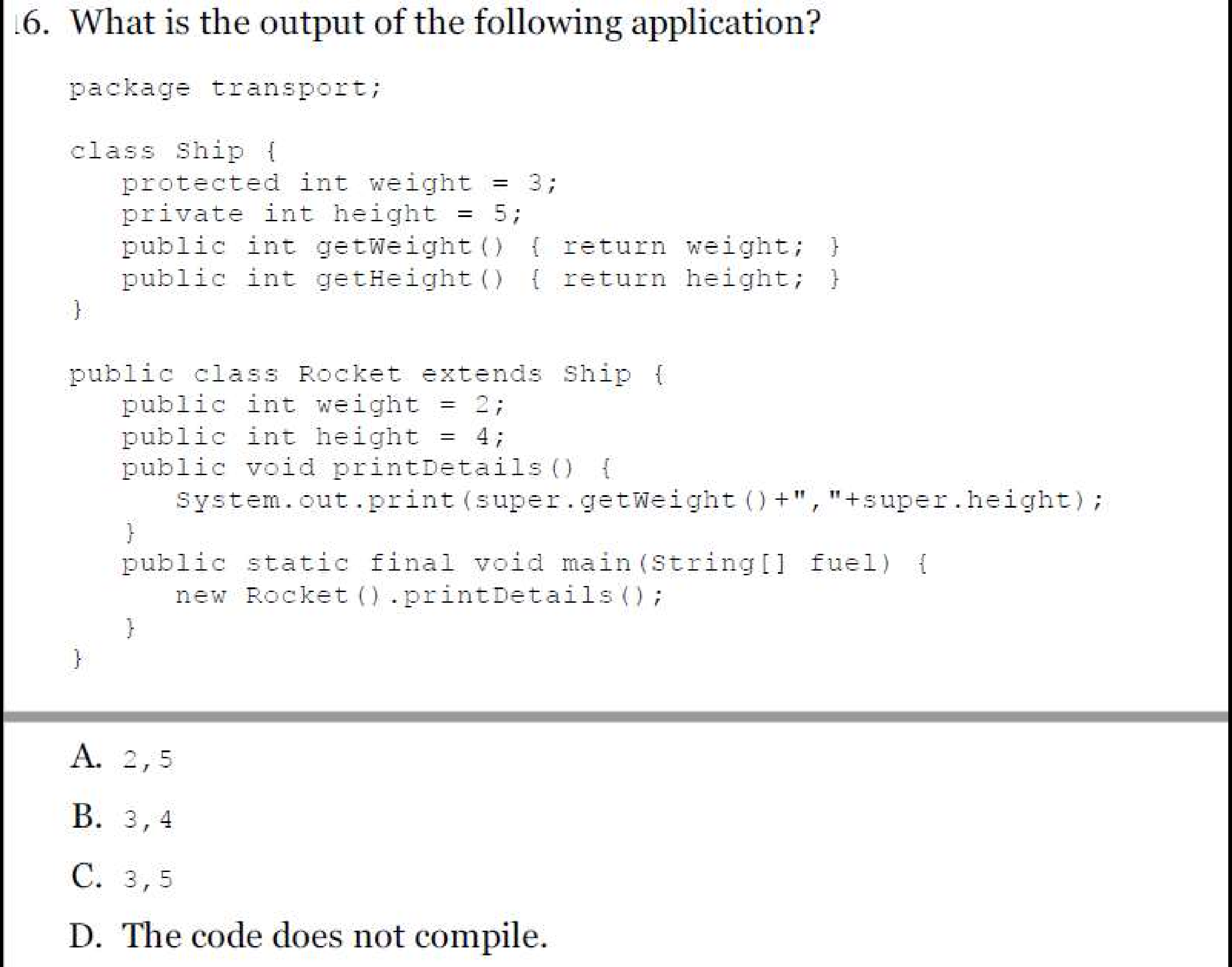
13. Which modifier can be applied to an interface method?

A. `protected`

B. `static`

C. `private`

D. `final`



16. What is the output of the following application?

```
package transport;

class Ship {
   protected int weight = 3;
   private int height = 5;
   public int getWeight() { return weight; }
   public int getHeight() { return height; }
}

public class Rocket extends Ship {
   public int weight = 2;
   public int height = 4;
   public void printDetails() {
      System.out.print(super.getWeight()+","+super.height);
   }
   public static final void main(String[] fuel) {
      new Rocket().printDetails();
   }
}
```

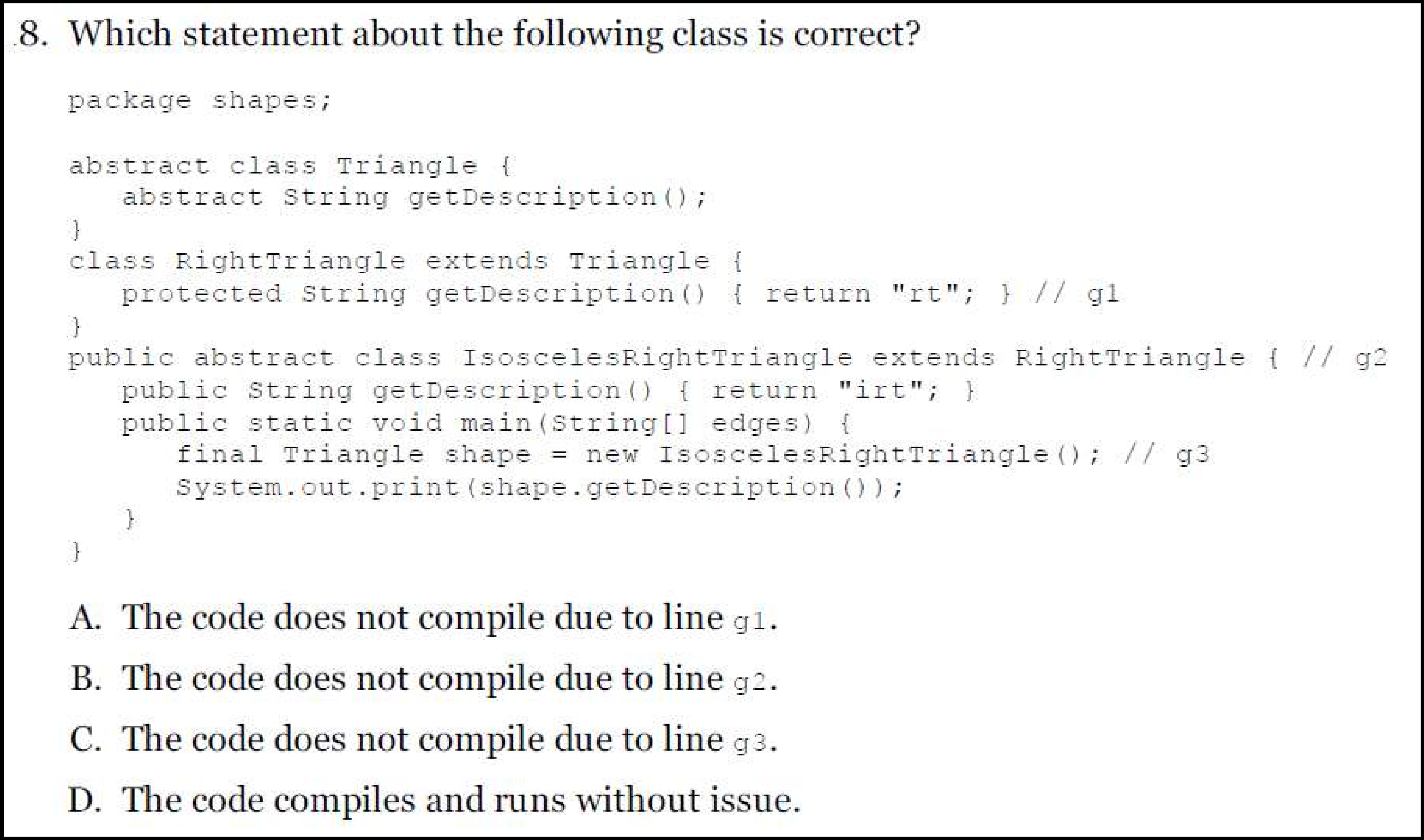
A. `2,5`

B. `3,4`

C. `3,5`

D. The code does not compile.

Deloitte.



8. Which statement about the following class is correct?

```
package shapes;

abstract class Triangle {
   abstract String getDescription();
}
class RightTriangle extends Triangle {
   protected String getDescription() { return "rt"; } // g1
}
public abstract class IsoscelesRightTriangle extends RightTriangle { // g2
   public String getDescription() { return "irt"; }
   public static void main(String[] edges) {
      final Triangle shape = new IsoscelesRightTriangle(); // g3
      System.out.print(shape.getDescription());
   }
}
```

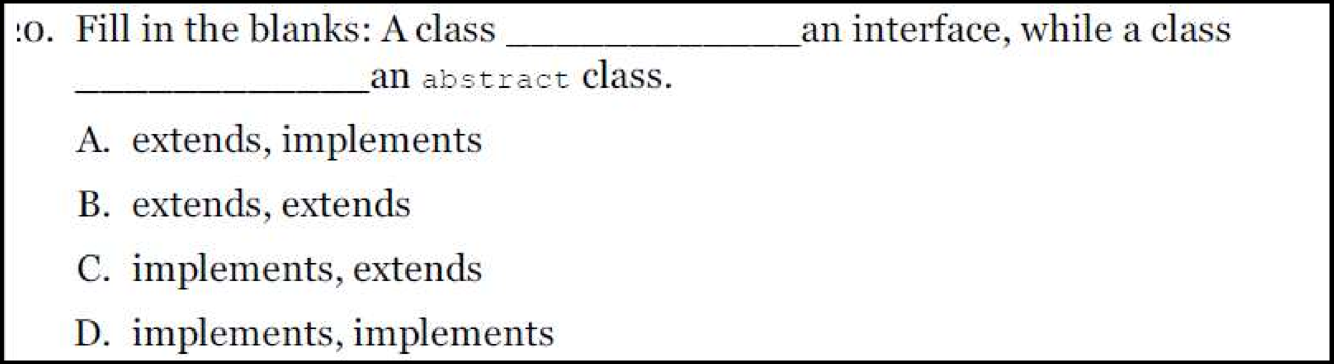
A. The code does not compile due to line `g1`.

B. The code does not compile due to line `g2`.

C. The code does not compile due to line `g3`.

D. The code compiles and runs without issue.

Deloitte.



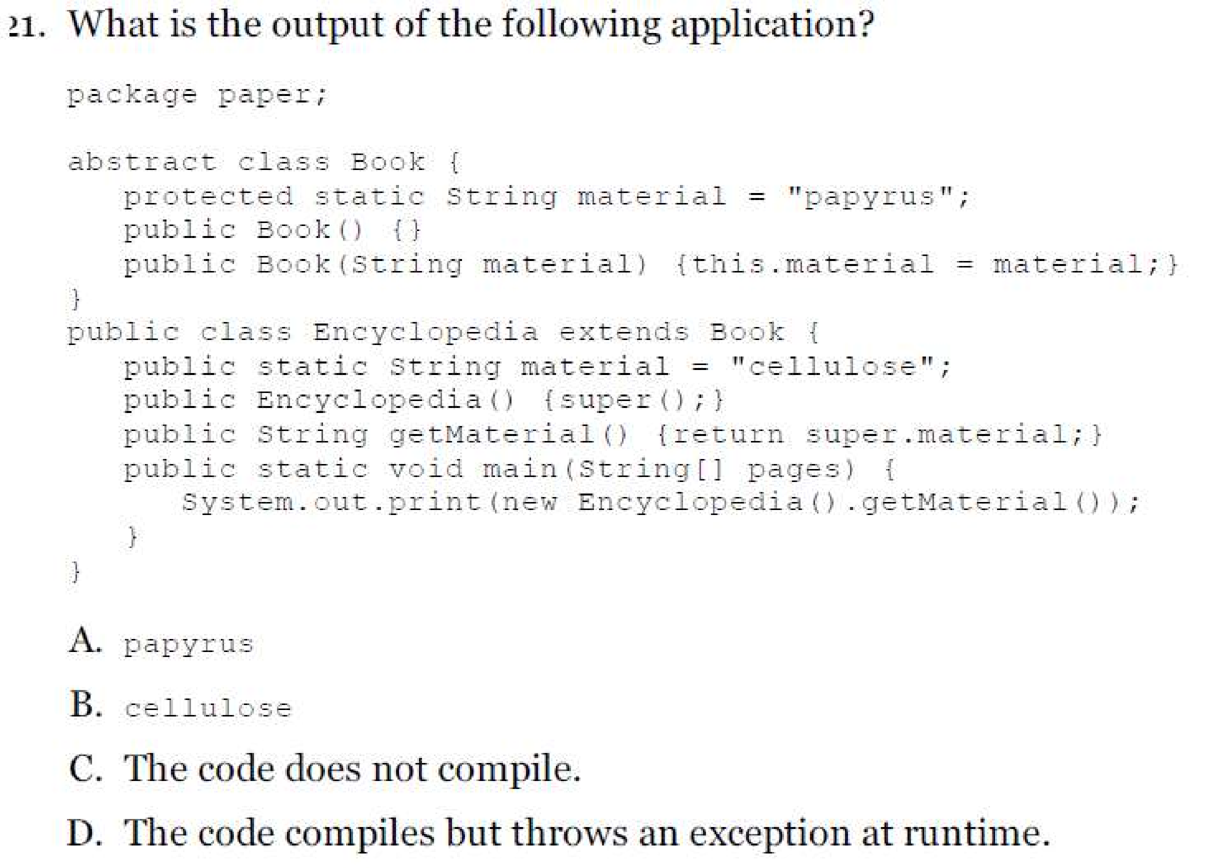
:0. Fill in the blanks: A class ________________an interface, while a class ________________an `abstract` class.

A. extends, implements

B. extends, extends

C. implements, extends

D. implements, implements



21. What is the output of the following application?

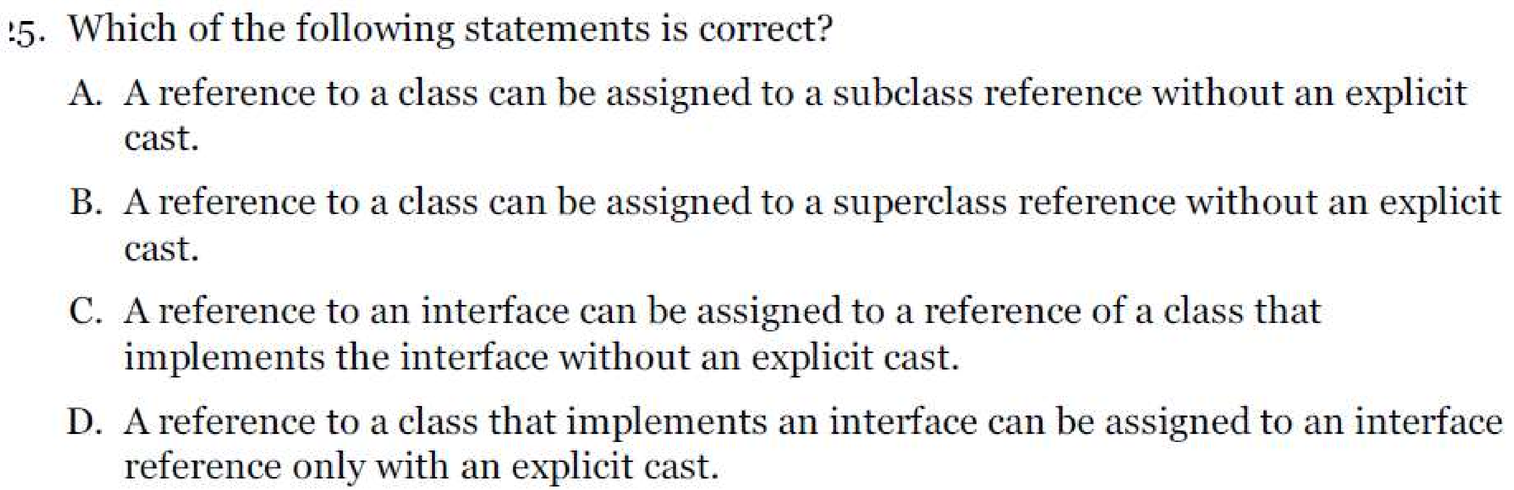
```
package paper;

abstract class Book {
   protected static String material = "papyrus";
   public Book() {}
   public Book(String material) {this.material = material;}
}
public class Encyclopedia extends Book {
   public static String material = "cellulose";
   public Encyclopedia() {super();}
   public String getMaterial() {return super.material;}
   public static void main(String[] pages) {
      System.out.print(new Encyclopedia().getMaterial());
   }
}
```

A. `papyrus`

B. `cellulose`

C. The code does not compile.

D. The code compiles but throws an exception at runtime.



:5. Which of the following statements is correct?

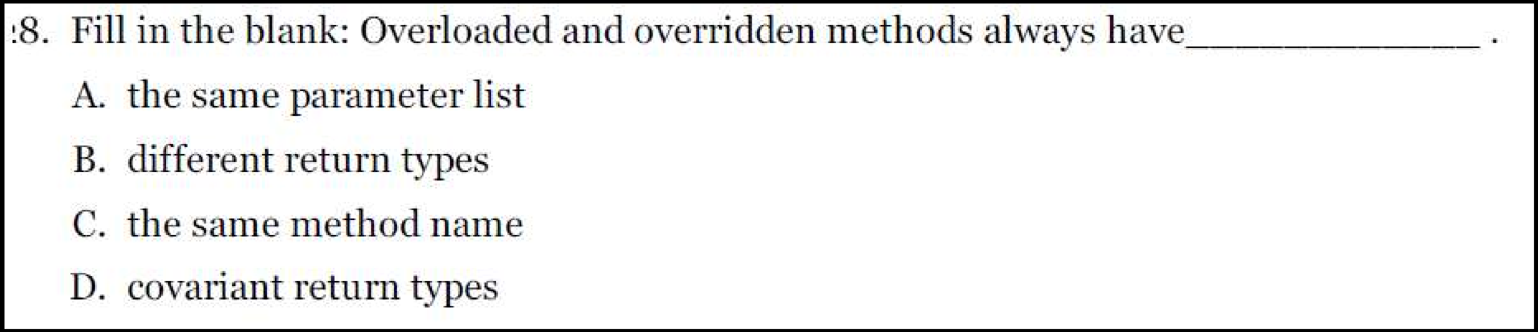
A. A reference to a class can be assigned to a subclass reference without an explicit cast.

B. A reference to a class can be assigned to a superclass reference without an explicit cast.

C. A reference to an interface can be assigned to a reference of a class that implements the interface without an explicit cast.

D. A reference to a class that implements an interface can be assigned to an interface reference only with an explicit cast.

Deloitte.



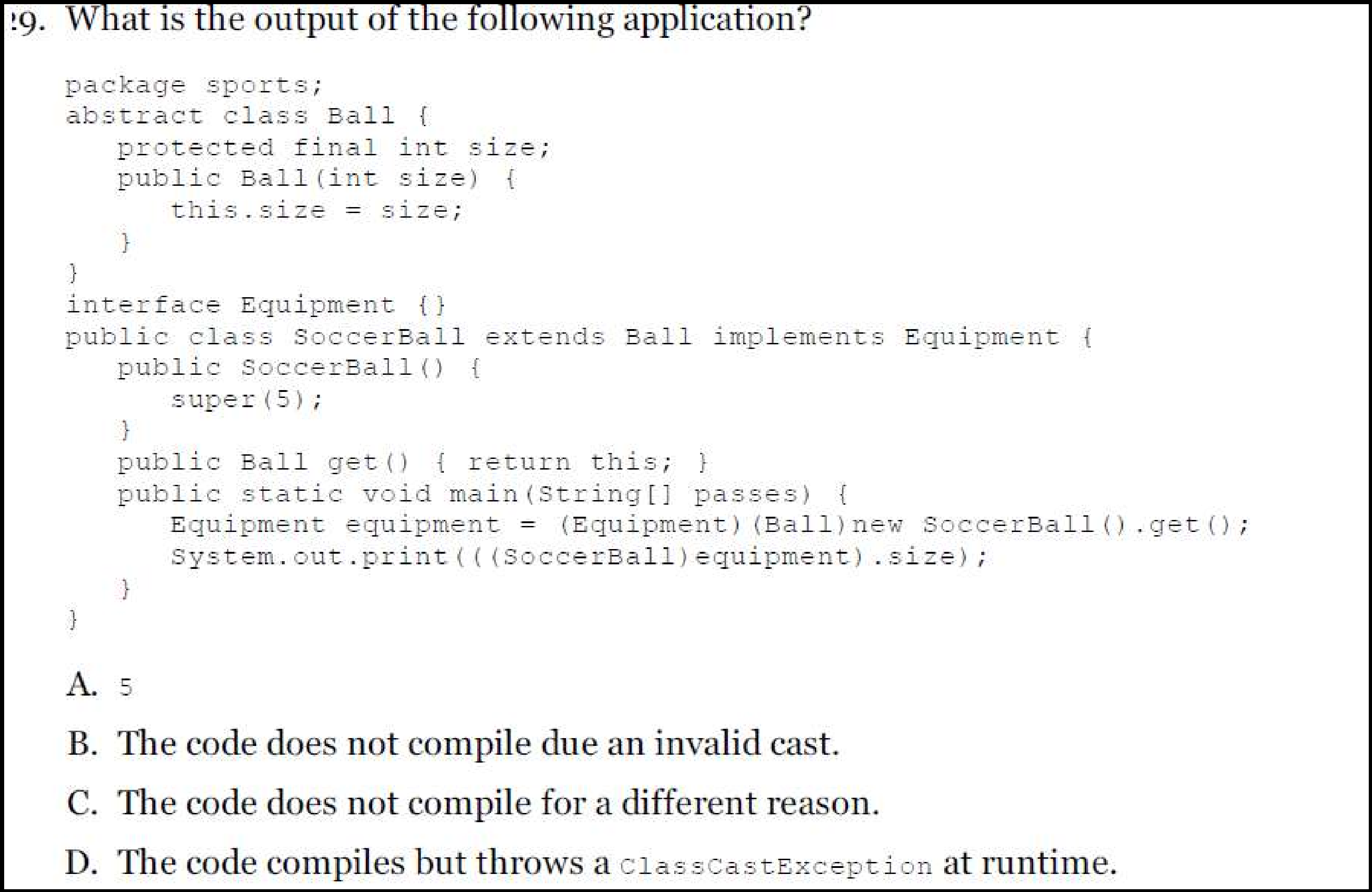
8. Fill in the blank: Overloaded and overridden methods always have______________ .

A. the same parameter list

B. different return types

C. the same method name

D. covariant return types



9. What is the output of the following application?

```
package sports;
abstract class Ball {
   protected final int size;
   public Ball(int size) {
      this.size = size;
   }
}
interface Equipment {}
public class SoccerBall extends Ball implements Equipment {
   public SoccerBall() {
      super(5);
   }
   public Ball get() { return this; }
   public static void main(String[] passes) {
      Equipment equipment = (Equipment)(Ball)new SoccerBall().get();
      System.out.print(((SoccerBall)equipment).size);
   }
}
```

A. `5`

B. The code does not compile due an invalid cast.

C. The code does not compile for a different reason.

D. The code compiles but throws a `ClassCastException` at runtime.