**Pregunta 1**

Given the following interface:

*public sealed interface Prints permits M1,M2{}*

What of the following definitions are correct?

1. sealed class M1 implements Prints{}

non-sealed interface M2 extends Prints{}

1. non-sealed interface M1 extends Prints{}

final class M2 implements Prints{}

1. final class M1 implements Prints{}

non-sealed interface M2 implements Prints{}

1. final interface M1 extends Prints{}

non-sealed interface M2 extends Prints{}

**Pregunta 2**

What is true about sealed class? (choose two)

1. sealed class and permits classes must be in the same module
2. sealed class and permits classes must be public
3. sealed class must not have a constructor
4. sealed class can extends another class
5. sealed class can be final

**Pregunta 3**

Given:

*interface Principal{*

*int id();*

*}*

*sealed class Person implements Principal permits Student, Teacher {*

*public int id(){*

*return 0;*

*}*

*}*

*record Student(int id, String subject) extends Person {}*

*sealed interface Teacher extends Person{}*

¿Why doesn't compile the above code? (choose 2)

1. Sealed class can't implement an interface
2. Sealed class must be declared abstract
3. A record can't extends a class
4. Teacher interface must specific a list of permitted interfaces/classes

**Pregunta 4**

Given:

*sealed class C1 { //Line 1*

*sealed class C2 extends C1 permits C3{ //Line 2*

*}*

*final class C3 extends C2{ //Line 3*

*}*

*class C4 extends C3{ //Line 4*

*}*

*void method() {*

*C1 c=new C3(); //Line 5*

*}*

*}*

Where is the compilation fail?:

A. Line 1

B. Line 2

C. Line 3

D. Line 4

E. Line 5