

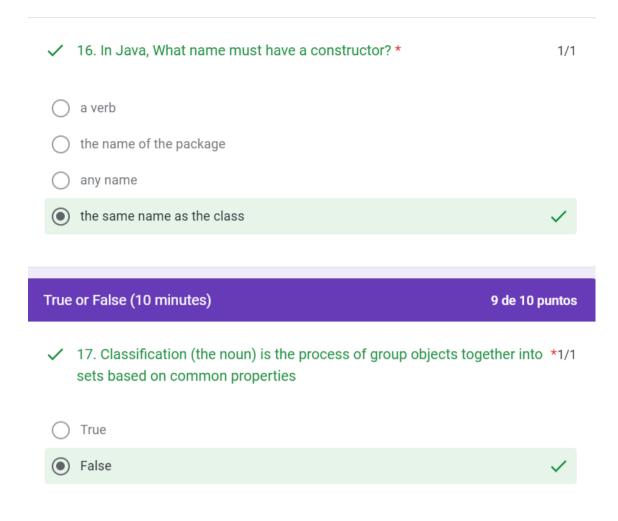
4. Leave unnecessary definitions out of the system implementation *	1/1	
Abstraction	✓	
Encapsulation		
Objects		
✓ 5. A class is composed of *	1/1	
Attributes and variables		
attributes and methods	~	
functions and methods		
★ 6. Inheritance in Object Orientation is also known as *		0/1
Generalization/Specialization		
Classification		×
Abstraction		
Respuesta correcta		
Generalization/Specialization		
 7. Reviews of software artifacts are of two types * 		1/1
O Unit tests and Desk checks		
O Inspections and unit tests		
Inspections and Walkthroughs		~

✓ 8. Encapsulation is implemented the keyword *	1/1
O public	
O void	
private	✓
 9. Aggregation, composition and association in a class diagram are relationships between classes that are implemented in code using 	*1/1
attributes	~
o methods	
O packages	
10. Dependency in a class diagram, is a relationship between classes th are implemented in	at *1/1
O variables	
Classes	
methods	~

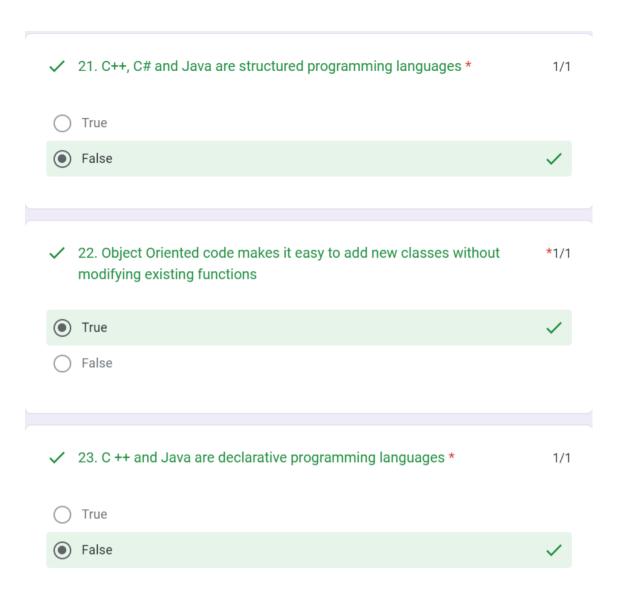
11. Match each of the following Git commands to its purpose *

	clone	add	commit	push	pull	Puntuación	
Put one or more new, changed, or deleted files under version control	0	•	0	0	0	1/1	~
Create a replica of remote repository into folder on the local system	•	0	0	0	0	1/1	✓
Make a new version	0	\circ	•	0	0	1/1	~
Sync any new versions on the local system with the remote system	0	0	0	0	•	1/1	✓
Sync any new versions on the remote system with the local system	0	0	0	•	0	1/1	✓

More OOP Fundamentals (5minutes)	5 de 5 puntos
12. What does it mean to instantiate a class object? *	1/1
O duplicate a class	
O delete a class	
create an object from the class	~
onnect two clases to each other	
✓ 13. A constructor is a *	1/1
variable	
Class	
attribute	
method	~
 14. These diagrams help organize and model the requirements of system showing the cases of use and actors? sequence diagrams collaboration diagrams 	fa *1/1
use case diagrams	✓
✓ 15. A getter is a *	1/1
method	✓
attribute	
class	



 18. Classification (the verb) or "class" is a set of objects that have the same kinds of attributes and methods True	he * 1/1
O True	
False	✓
19. One way to find potential classes in a system is to document a helevel description of the system and look for nouns. Those nouns are most likely to represent meaningful classes.	
True	~
○ False	
20. Use cases are used to document the requirements (the goals) o system	fa * 1/1
True	~
○ False	



X 24. A method that is called from another method inside the same class should be defined later in the same class, i.e., after the method that called × True False Respuesta correcta False 25. WheelsList is a good name for a variable * 1/1 True False ✓ 26. A good programming practice is to use nouns to name the methods * 1/1 True False Otro:

X 27. Classes + Relations (Reverse Engineering). Draw the class diagram *.../10 corresponding to the following code. Convert every attribute to associations, aggregations, compositions or dependencies, with appropriate names and multiplicity constraints (Upload the Png/Jpg file here, and the vpp and png file to the repository):

```
Pauthor OOP instructors */
         c class POOExam1P27 {
      public static void main(String[] args) {
        E e = new E();Question10
        Cc = new C();
        e.m2(c);
        //Optional code
        A = new A();
        B b1 = new B();
        B b2 = new B();
        a.m1(b1, b2);
      }
    public class A {
      private B b1;
      private B b2;
       * This method uses two objects of type B
       * @param x of type B
       * @param y of type B
      public void m1(B x, B y){
      }
    public class B {
    public class C {
      private B b;
public class E {
  private A a;
  private B[] b = new B[10];
  /**
  * This method will allow to use an object of type C, and returns nothing
  * @param z this is an object of type C
  */
  public void m2(Cz) {
```

Based on the previous answer (Class diagram). Answer the following questions with an integer number. use digits. DON'T use words

