Object-oriented Programming

By Arantza Beitia

THEORICAL PART

1. What is **object-oriented programming** in general terms?
2. What is a **class**?
3. What is an **object**?
4. What is an **instance**?
5. What is a **property**?
6. What is a **method**?
7. What is the difference between a **function** and a **method**?
8. What is a **constructor**?
9. What is the difference between a **class**, an **object** and an **instance**?
10. What do we understand about the concept of **encapsulation**?
11. What do we understand about the concept of **abstraction**?
12. What do we understand about the concept of **inheritance**?
13. What do we understand about the concept of **polymorphism**?
14. What do we understand about the concept of **Overload**?
15. What do we understand about the concept of **Override**?
16. What differences exist between the concept of **Overload** and **Override**?
17. What is a **static class**?
18. Look for 3 advantages over **object-oriented programming** compared to other programming paradigms
19. Look for **disadvantages of this paradigm**.

01

What is **object-oriented programming** in general terms?

Is a program design paradigm.

02

What is a **class**?

A class is a template with the purpose of creating many objects.

03

What is an **object**?

An instance of a class.

04

What is an **instance**?

The object is a copy of the class. The instance is a variable that contains the memory address of the object. There can be multiple objects of the same class and then multiple instances of each of those objects.

05

What is a **property**?

Characteristics of a class (attributes)

06

What is a **method**?

Actions of a class

07

What is the difference between a **function** and a **method**?

They only differ in that the methods are used inside the classes and by their special declaration.

08

What is a **constructor**?

Is a method to initialize an object

09

What is the difference between a **class**, an **object** and an **instance**?

Class - template
Object - Instance of a class
Instance - Copy of an object

10

What do we understand about the concept of **encapsulation**?

Is a protection mechanism for the data and methods inside the class to restricting the access, with the purpose of making the code more secure and robust.

11

What do we understand about the concept of **abstraction**?

An abstract class can't be instantiated, only inherited by another class.

What do we understand about the concept of **inheritance**?

Is the way of sharing properties and methods between classes and objects.

13

What do we understand about the concept of **polymorphism**?

Is the mechanism by which we can “relax the type system”, so that it also accepts objects from child or derived classes.

What do we understand about the concept of **Overload**?

Is the ability to create multiple functions of the same name with different implementations.

15

What do we understand about the concept of **Override**?

It is used to replace parent method in child class. The purpose of overriding is to change the behavior of parent class method.

16

What differences exist between the concept of **Overload** and **Override**?

Overload changes the parameters of the same method, but doesn’t changes the method as Override.

17

What is a **static class**?

A method accessible without an instantiation of the class. It doesn't have to be part of an object.

Look for 3 advantages over **object-oriented programming** compared to **other programming paradigms**

- Reusable: good design of the classes allods us to reuse them in different parts of the program.
- Maintainable: easier to reed and to understand.
- Scalable: The ease of adding, deleting or modifying new objects allows us to make modifications in a very simple way.

Look for disadvantages of this paradigm.

- Change in way of thinking.
- Execution is slower.
- The need to use class libraries forces their learning and training.