10. Difference between a Constructor and a Method?

- Constructor doesn't have a return type and constructor's name must be same as the class name.
  - Constructor is called automatically when a new object is created. Constructor is invoked implicitly.
  - The Java compiler provides a default constructor if we don't have any constructor.
  - Constructors are not inherited by child classes
- Method have a return and the method's name may or not be same as the class name
  - Method is invoked explicitly.
  - Method is not provided by compiler in any case.
  - Methods are inherited by child classes.

11. What is the difference between a local variable and an instance variable?

- A **local variable** is typically used inside a method, constructor, or a block and has only local scope. Thus, this variable can be used only within the scope of a block.
- The best benefit of having a local variable is that other methods in the class won't be even aware of that variable.
  Example

```
if(x > 100){
    String test = "Alberto";
}
```

- An **instance variable** in Java, is a variable which is bounded to its object itself. These variables are declared within a class, but outside a method. Every object of that class will create its own copy of the variable while using it. Thus, any changes made to the variable won't reflect in any other instances of that class and will be bound to that particular instance only.
  Example

```
class Test{
    public String EmpName;
    public int empAge;
}
```

12. Object Oriented Programming (OOP)

- OOP is a programming language model organized around object rather than actions (logic and functions).
- In other words, OOP mainly focuses on the objects that are required to be manipulated instead of logic. This approach is ideal for the programs large and complex codes and needs to be actively updated or maintained.;
- It makes development and maintenance easier - It provides data hiding - It provides ability to simulate real-world.

OOP language follow 4 principles:

- **Encapsulation** : We can hide direct access to data by using private key and we can access private data by using getter and setter method.
- **Abstraction** : It is a process of hiding implementation details and showing only functionality to the user. Abstraction lets you focus on what the object does instead of how it does it.
- **Inheritance** : It is used to define the relationship between two classes. When a child class acquires all properties and behaviors of parent class known as inheritance. Child class can reuse all the codes written in parent class. It provides the code reusability.
- **Polymorphism** : It is an ability of object to behave in multiple form. The most common use of polymorphism in Java is when a parent class reference type of variable is used to refer to a child class object.

  Example

```
WebDriver driver = new ChromeDriver();
```

  We use method overloading and overriding to achieve Polymorphism.