How OOP's concepts in Test Automation are useful ?

Below are just few examples:  
  
• Class-Object: The heart of OOP. E.g. you create Page Objects in the Test classes to perform relevant functions.  
• Encapsulation: Access web elements via Page Object methods. No direct access.  
• Abstraction: From the Test classes, just call the Page class methods to achieve the required functional flows.  
• Inheritance: Most of the frameworks have a Base Page class and a Base Test class which is extended in all other page & test classes. Generally used for setup & initialization.  
• Polymorphism: There are times when we need methods differing in the arguments to handle different flows.

## Q. Selenium Advantages and Limitations

A. “Selenium”, the de-facto top automation testing tool.

• Open source testing tool.

• Capability to operate on almost every OS.

• Supports multiple languages.

• Independent of the language that the web application is using.

• Supports a range of browsers.

• Very dynamic developer & helping community and user base.

• Robust element locators.

• Convenient to implement frameworks that revolve around OOP.

• Support for integration of open source frameworks like TestNG + Version control using Maven, Jenkins.

• Execute simultaneous tests leveraging various browsers on various machines.

• Supports Web and Mobile Web Applications.

**Few Limitations:**

• No dedicated official technical support.

• Supports only Web based applications.

• Dependence on third-party tools for complete benefit, unlike proprietary all-in-one tools.

• Limited support for Image-based Testing, Captcha and Bar code readers.

• Unstable new releases. It evolves with time.

• No implicit Test Tool integration for Test Management.

• No Built-in Reporting facility.

• Lot of challenges with IE browser.

Being an open-source tool, these limitations are okay to live with.

Q. What’s POM.xml?

A. Project Object Model: an XML file that contains information about the project and configuration details used by Maven to build the project.

In the normal project, you will add JAR files and libraries as required. In Maven-based project, those JAR files, libraries are added to the project using this pom.xml. In the POM context we call those JAR files & libraries as ‘dependencies’.

Some of the configuration that can be specified in the POM are the project dependencies, the plugins or goals that can be executed, the build profiles, and so on.

If there are enough entries in the pom.xml then that is all you need! Import it to Eclipse >> Maven will download your source code from CVS, download various dependency jars (like log4j, Apache Commons, json, etc.), run the tests, build the jar/war, deploy to your app server, generate a report, etc. Each task is mentioned as a goal.

The most popular use: when there is no Maven, it needs to add all the library JAR files one by one to the project. But when there is Pom.xml there is no need to add library JAR files one by one. Simply add the dependency to the Pom.xml, and it will automatically add the library JAR files to the project.

In short the pom.xml will have all information to build your project.

BDD??

|  |
| --- |
|  |

Behavior-driven development (BDD) is a software development methodology in which an application is specified and designed by describing how its behavior should appear to an outside observer. It’s an extension of TDD. Like TDD, in BDD also we write tests first and then add application code. The major difference is that tests are written in plain descriptive English type grammar and are explained as behavior of application and are more user focused. This difference brings in the need to have a language which can define requirements in an understandable format.

BDD offers collaboration between Business stakeholders, Business Analysts, QA Team and developers using ubiquitous language which is easy to describe. It extends Test Driven Development (TDD) by utilizing natural language that non-technical stakeholders can understand.

BDD frameworks such as Cucumber or JBehave are an enabler, acting a “bridge” between Business & Technical Language.

## Q.We have a table with employees’ data, write a query to reverse the gender column value. Need to update it in a single query.

## Answer: Update EmployeeTable set "gender" = (case "gender" when 'male' then 'female' else 'male' end);