# 1. What is Maven and why it is used?

Answer:  
During building the [software system](https://www.educba.com/what-is-application-software-and-system-software/), you would like to perform the subsequent tasks no matter which event methodology you have used.  
•Gather necessities  
•Analyse necessities  
•[Design](https://www.educba.com/course/advanced-database-designer-4/) solution  
•Develop (code) solution  
•Test solution  
•Deploy solution  
•[Maintain solution](https://www.educba.com/how-to-maintain-a-good-work-life-balance/)

The above tasks are usually executed in the order listed above. For example, you cannot develop a solution before you analyze the requirements and design it. Some may argue that with the Test Driven Development (Book), the test cases are created before the solution is developed. While that is true, the tests are still executed after the development is done irrespective of when the tests were developed.

While all the thinking should be done by the team, the repetitive steps may be disbursed by the associate assistant, if we may. The assistant isn’t intelligent however is aware of the software system development flow and the way to try to staple items. For instance, the assistant will produce the comes supported templates, transfer needed third-party libraries (also referred to as JARs) and their dependencies too, build all elements and build a deployable JAR, WAR or EAR file. This assistant is termed, maven. This distinguishes[maven from alternative building tools](https://www.educba.com/maven-commands/) like hymenopteran (Homepage) or alternative build scripts.

In another word, maven is a project [management tool](https://www.educba.com/best-database-management-tools/). Maven provides developers a whole build [lifecycle](https://www.educba.com/course/warranty-lifecycle-management/)framework which can be used by the development team that will change the[project](https://www.educba.com/course/project-finance-tutorials/)‘s infrastructure as maven uses a default build lifecycle.

# 2. What Are the Aspects Maven Managed?

Answer:  
Maven provides developers ways in which to manage following:  
•Builds  
•Documentation  
•[Reporting](https://www.educba.com/course/financial-reporting-analysis-cfa-level-2-training/)  
•Dependencies  
•SCMs  
•Releases  
•Distribution  
•mailing list

# 3. What Is Pom? What Information Does Pom Contain?

Answer:  
POM stands for Project Object Model. It’s basic Unit of work in maven. it’s an [XML file](https://www.educba.com/course/xml-1/). It perpetually resides within the base directory of the project as pom.xml. To create the project(s), it contains numerous configuration details and data concerning the project employed by maven  
POM contains the some of the following configuration information −  
•project dependencies.  
•plugins.  
•[goals](https://www.educba.com/what-are-your-career-goals/).  
•build profiles.  
•project version.  
•developers.  
•mailing list.

# 4. What Is Maven Build Lifecycle? Name The 3 Build Lifecycle of Maven?

Answer:  
A Build Lifecycle is a well-defined sequence of phases that outline the order in which the goals are to be executed. Here phase represents a stage in life cycle.

The three build lifecycles are:

•Clean: cleans up artifacts created by previous builds.  
•Default (or build): this can be accustomed to build the appliance.  
•Site: generates site documentation for the project.

# 5. What Is Maven Artifact?

Answer:  
An artifact is a file, which is internally a JAR file that gets deployed to a [Maven repository](https://www.educba.com/install-maven/). A Maven build produces one or more artifacts, such as “sources” JAR and a compiled JAR.

Each artifact encompasses a group ID (usually a reversed domain name, like com.example.foo), an artifact ID (just a name), and a version string. The 3 along unambiguously determine the artifact. A project’s dependencies are specified as artifacts.

# 6. What Are the Phases of a Maven Build Lifecycle?

Answer:  
Some major principles of [agile testing](https://www.educba.com/course/agile-mobile-testing/) are:

Following are the phases: −

• Validate − validate the project is correct and all necessary information is offered.  
• Compile − compile the source code of the project.  
• Test − test the compiled source code employing an appropriate unit testing framework and these tests should not require the code deployed or packaged.  
• Package − take the compiled code and package it in its distributable format like a JAR.  
• Integration-test − process and deploy the package if necessary into an n atmosphere wherever integration tests may be run.  
• Verify − run any tests to verify the package is still valid and meets quality requirements.  
• Install − Install the package into the native repository, to be used as a dependency in alternative projects regionally.  
• Deploy − copies the final package to the remote repository for sharing with alternative projects and developers. It is done in an [integration](https://www.educba.com/course/project-management-2/) or release environment,

# 7. What is a Maven Repository?

Answer:  
A Maven repository is a location where all the plugins, library, jars, project jars or any other particular project related information is stored and can be easily used by Maven.  
Their types are 3 types of maven repository:  
•local repository  
•central repository  
•remote repository

# 8. What is Maven snapshot?

Answer:  
A snapshot version of Maven is not a real version and that will not have been released. An equivalent version might get several updates. Usually, exposure dependencies ought to solely exist throughout development and no free version (non-snapshot) ought to have a dependency on a snapshot version.

# 10. What features should a build tool have?

Answer:  
A build tool should have:

* Compile [Java code](https://www.educba.com/course/java/) and build jar, war and ear files for deployment and release.
* Versioning and dependency management
* Run take a look at and report test results
* Run code quality check with Sonar, check style, Find bugs, etc.
* Environment property substitution
* Files generations ([WSDL](https://www.educba.com/course/wsdl-training/), AspectJ, XSL, etc.)
* Support for cross-platform (UNIX Windows) and IDEs (Eclipse, Netbeans, IntelliJ)
* Proper documentation and support.

# 11. List out the dependency scope in Maven?

# The various dependency scope employed in maven are:

# **Compile**: it’s the default scope, and it indicates what dependency is offered within the classpath of the project.

# **Provided**: It indicates that the dependency is provided by JDK or internet server or container at runtime.

# **Runtime**: This tells that the dependency isn’t required for compilation, however, is needed throughout the execution.

# **Test**: It says dependency is offered just for the take a look at compilation and execution phases.

# **System**: It indicates you have got to give the system path.

# **Import**: this means that the known or nominative pom ought to get replaced with the dependencies in this POM’s section