Research about the Maven build tool:

Maven is a popular build automation and project management tool used primarily for Java projects. It simplifies the build process by providing a standard structure for projects, managing project dependencies, and automating tasks such as compilation, testing, packaging, and documentation.

Here are some key aspects and features of Maven:

1. Project Object Model (POM): Maven uses a Project Object Model, defined in an XML file called `pom.xml`. This file describes the project's configuration, dependencies, plugins, and other settings. The POM provides a standardized way to manage projects and helps Maven understand how to build the project.

2. Dependency Management: Maven simplifies dependency management by allowing you to declare project dependencies in the `pom.xml` file. It can automatically download the required dependencies from a central repository like Maven Central, making it easier to manage and share libraries.

3. Build Lifecycle: Maven follows a predefined build lifecycle, consisting of phases such as `clean`, `compile`, `test`, `package`, `install`, and `deploy`. Each phase represents a specific step in the build process. Developers can execute these phases to perform specific tasks.

4. Plugins: Maven uses plugins to extend its functionality. Plugins provide additional goals that can be executed during different phases of the build

lifecycle. There are plugins available for various tasks, such as compiling code, running tests, generating documentation, and more.

5. Central Repository: Maven Central Repository is a centralized repository for storing and distributing project dependencies. When you declare dependencies in your `pom.xml` file, Maven automatically downloads the required JAR files from this repository.

6. Convention over Configuration: Maven follows the principle of "convention over configuration," which means that it relies on a standard project structure and default configurations. This reduces the need for extensive build scripts, as Maven can infer many settings from the project structure.

7. Transitive Dependencies: Maven handles transitive dependencies, meaning it automatically includes dependencies of your project's dependencies. This simplifies the process of managing a project's dependencies, as you only need to specify direct dependencies.

8. Multi-module Projects: Maven supports multi-module projects, allowing you to break a large project into smaller, more manageable modules. Each module can have its own POM file, and Maven can build the entire project or specific modules.

9. IDE Integration: Maven integrates well with various Integrated Development Environments (IDEs) such as Eclipse, IntelliJ IDEA, and NetBeans, making it easier for developers to work with Maven projects.

10. Command-Line Interface: Maven can be used from the command line, making it suitable for automated build processes and continuous integration systems.