1,What is the purpose of the core module in AEM?
The core module in an AEM project contains all the backend logic, mainly written in Java. It serves as the backbone for dynamic components by providing Sling Models, OSGi services, Servlets, and business logic.

2, What kind of files and code can be found in the core folder?

1. Sling Models (@Model)
2. OSGi Services
3. Servlets
4. Helper Utilities
5. pom.xml

3, Explain the role of ui.apps in AEM projects.
The ui.apps module contains all the frontend and content-related parts of the project. It includes:

- HTL files
- Dialogs and Design Dialogs
- CSS, JS, and ClientLibs
- Templates and Page Components
- Content structure (JCR nodes)

4, How are components structured in the ui.apps folder?
Each component has its own folder inside /apps/project/components.
Every component consists of:

1. HTL (helloWorld.html)
2. Dialogs (_cq_dialog.xml)
3. Edit Config (_cq_editConfig.xml)
4. .content.xml

5, Hello World Component:

- **Where is the Hello World component located in both core and ui.apps?**
  Core: core/src/main/java/com/project/core/models/HelloWorldModel.java

  ui.apps: ui.apps/src/main/content/jcr_root/apps/project/components/helloWorld
- **Explain the Java class (in core) for the Hello World component.**

  This Sling Model fetches a title from JCR and provides a method to get it:

- **How does the HTL script work in ui.apps for Hello World?**
  The HTL script in ui.apps for the Hello World component retrieves data from the Sling Model and renders it dynamically. It uses expression language (e.g., ${model.title}) to

access model properties and ensures automatic escaping for security. This approach keeps the markup clean and separates logic from presentation.

- **How are properties and dialogs defined for this component?**
  _cq_dialog.xml → Defines an input field for the title:

6, What are the different types of AEM modules (core, ui.apps, ui.content, etc.)?
four types:

1,core

2,ui.apps

3,ui.content

4,all

7,How does Maven build these modules?
Maven processes the modules using pom.xml.

- **core** → Builds an OSGi bundle.
- **ui.apps** → Creates a content package using content-package-maven-plugin.
- **all** → Merges everything into a deployable AEM package.

8, Explain the build lifecycle of Maven in the context of AEM.
1, **Clean** → Deletes previous build files.

2, **Compile** → Converts Java code into bytecode.

3, **Package** → Creates JAR (core) or ZIP (ui.apps).

4. **Install** → Installs built artifacts locally.

5, **Deploy** → Installs into AEM.

9, How are dependencies managed in pom.xml?
Maven fetches dependencies from **Adobe's Maven repository**.

10, Why is Maven used instead of other build tools?
1,Native support for multi-module projects

2,OSGi-friendly dependency resolution

3,AEM archetype support

4,Automation for package deployment

11, What advantages does Maven offer for AEM development?

- Standardized project structure
- Handles OSGi dependencies
- Automates package creation and deployment
- Supports different environments using profiles

12, How does Maven help in managing dependencies and plugins in AEM projects?
Uses dependency management to fetch necessary AEM APIs.

Uses plugins for package creation, deployment, and Sling models.

13, What does mvn clean install do in an AEM project?
1,Cleans previous builds

2,Compiles code

3,Packages modules

4,Installs them locally

Deploys to AEM (if using autoInstallPackage)

14, How to deploy packages directly to AEM using Maven commands?
mvn clean install -PautoInstallPackage

15, Explain the purpose of different Maven profiles in AEM (autoInstallPackage, autoInstallBundle).
Maven profiles in AEM define different build and deployment configurations based on project needs. The **autoInstallPackage** profile deploys the full content package, while **autoInstallBundle** installs only the OSGi bundle. The **integrationTests** profile is used to run tests against the AEM instance. Additionally, profiles like **dispatcherConfig** help deploy dispatcher configurations, and **author** or **publish** profiles target specific AEM environments. These profiles streamline the build process and enable flexible deployments.

16, What is the purpose of dumplibs in AEM?
It helps debug **Client Libraries (CSS/JS)** loading issues.

17, How can you view client libraries using dumplibs?
Go to:

ruby

CopyEdit

http://localhost:4502/libs/granite/ui/content/dumplibs.html

18, Explain how client libraries are structured in AEM.
Located in:

bash

CopyEdit

/apps/project/clientlibs

Contains:

- css.txt - Lists CSS files.
- js.txt- Lists JS files.
- categories - Defines where it's used.