Module – 1

Flutter – Introduction

Q. What is flutter?

Ans. Flutter is a mobile app development framework created by Google. It allows developers to build high-quality, native mobile apps for both iOS and Android using a single codebase. Flutter uses a programming language called Dart, which is easy to learn and has a syntax similar to Java and JavaScript.

One of the key benefits of Flutter is its ability to create beautiful, responsive user interfaces using a library of pre-built widgets. These widgets can be customized to fit the specific needs of your app, and can be animated and styled to create a unique look and feel.

Flutter also has a hot reload feature, which allows developers to see changes to their code in real-time without having to rebuild the entire app. This makes the development process faster and more efficient.

Overall, Flutter is a powerful and flexible framework that is great for building mobile apps of all kinds. Whether you're building a simple app or a complex one, Flutter has the tools and resources you need to get the job done.

Q. Why did you choose flutter?

Ans. Flutter is a great choice for mobile app development for a number of reasons. Here are a few:

1. Fast development: Flutter's hot reload feature allows developers to see changes to their code in real-time without having to rebuild the entire app. This makes the development process faster and more efficient.

2. Single codebase: Flutter allows developers to build high-quality, native mobile apps for both iOS and Android using a single codebase. This means that developers can save time and resources by not having to write separate code for each platform.

3. Beautiful UI: Flutter uses a library of pre-built widgets that can be customized and styled to create beautiful, responsive user interfaces. This makes it easy to create a unique look and feel for your app.

4. Large community: Flutter has a large and supportive community, which means there are plenty of resources available for learning and troubleshooting. This makes it easier for developers to get started with Flutter and to find help when they need it.

Overall, Flutter is a powerful and flexible framework that is great for building mobile apps of all kinds. Whether you're building a simple app or a complex one, Flutter has the tools and resources you need to get the job done.

Q. What is dart?

Ans. Dart is a programming language created by Google. It is used primarily for building web and mobile applications, as well as server-side applications. Dart is an object-oriented language that is very similar to Java and C#, making it easy for developers who are familiar with those languages to learn.

One of the key features of Dart is its support for both just-in-time (JIT) and ahead-of-time (AOT) compilation. This allows developers to write code that can be compiled and run quickly, making it ideal for building fast, responsive applications.

Dart also has a number of built-in features that make it easy to write clean, maintainable code. For example, Dart has a strong type system that helps catch errors at compile-time, rather than at runtime. Dart also supports asynchronous programming, which makes it easy to write code that can run in the background without blocking the main thread.

Overall, Dart is a powerful and flexible language that is great for building a wide range of applications. Whether you're building a web app, mobile app, or server-side application, Dart has the tools and resources you need to get the job done.

Q. Difference between hybrid application and native application

Ans.

No.	Categories	Hybrid Application	Native Application
1.	**Development**	Hybrid applications rely on a web view to render content, which can be slower and less responsive.	Native applications generally have better performance than hybrid applications because they can take full advantage of the hardware and software features of the device.
2.	**Performance**	Hybrid applications rely on a web view to render content, which can be slower and less responsive.	Native applications generally have better performance than hybrid applications because they can take full advantage of the hardware and software features of the device.
3.	**User experience**	Hybrid applications can feel less native and responsive because they are essentially web applications running inside a native container.	Native applications provide a better user experience because they are designed specifically for the platform they are running on and can take advantage of platform-specific features like gestures, animations, and notifications.
4.	**Maintenance**	Hybrid applications have a single codebase that can be used for both platforms, which can make maintenance easier and more efficient.	Native applications require separate codebases for each platform, which can make maintenance more difficult and time-consuming.

5.	**Cost**	Hybrid applications can be less expensive to develop because they have a single codebase that can be used for both platforms and rely on web technologies that are more widely known	Native applications can be more expensive to develop because they require separate codebases for each platform and specialized programming skills for each platform.
6.	**Time to market**	Hybrid applications can be developed more quickly than native applications because they have a single codebase that can be used for both platforms.	Native applications take longer to develop because they require separate codebases for each platform.

Q. Full Form of SDK, APK, ADB, AVM

Ans.

1. SDK - Software Development Kit

2. APK - Android Application Package

3. ADB - Android Debug Bridge

4. AVM - Android Virtual Machine