Kotlin
The Programming Language For Android

Srikanth Reddy

Android Developer, Technical Trainer KMIT

Course Syllabus

- Kotlin fundamentals
  - Kotlin Introduction, Basics and functions
  - OOP with classes and objects
  - Generics
  - Functional Manipulation
- Introduction to Android app development SDK.
- Layouts, styles and Design
  - Linear layout using the Layout Editor
  - Add user interactivity
  - Constraint layout using the Layout Editor
  - Data-binding basics
  - Styles and themes
  - Material Design, dimens, and colors
- Navigation
  - Create a fragment
  - Define navigation paths
  - Start an external activity

- Activity and fragment lifecycles
- Architecture components, Connecting to database, Room DB and coroutines
- RecyclerView and Connecting to the internet
- Notifications and Login with FireBaseUI.
- Advanced Graphics and Animation
- Geo-Mapping

Course Outcome

- Learn the core Kotlin skills
- Learn the core Android app development environment.
- Understand how to create Android apps using Kotlin.
- You'll learn to use Android components Activity, Fragments, Handlers, Services, Room for databases, Work Manager for background processing, the new Navigation component, and more.
- You'll use key Kotlin features to write your app code more quickly and concisely.

Week 1 : Session 1

Kotlin Introduction, History, Installation and Program

Agenda

- Introduction and Background
- What is Kotlin?
- History
- Benefits of Kotlin
- Task: Install the Java Development Kit (JDK)
- Task: Install IntelliJ IDEA
- Task: Create Hello Kotlin

Kotlin Introduction and Background

- Kotlin is modern programming language.
- The basics of a modern, object-oriented, statically typed programming language such as Java or C#
- Since 2017, Google has officially supported Kotlin for developing Android apps.
- Apple got a nice(r) new language(Swift), Android stuck with Java(Missing modern features)- Lambdas, properties and handling null values.
- Features specific to JDK (and Android API)

What is Kotlin?

- Named after island in St. Petersburg
- Programming language
  - Based on the JVM
  - Compact and modern (“better Java”)
  - Open source
- Created by JetBrains
  - Built into Android Studio and IntelliJ IDEA
  - Used by JetBrains internally

History

- Project Kotlin unveiled in July 2011
- Kotlin 1.0 released in February 2016
- “Language of the Month” - Dr. Dobb’s Journal (01/2012)

Benefits of Kotlin

- It's focused on clarity, conciseness, and code safety
- Robust code : Kotlin is strongly typed
  - null-pointer exceptions in software have caused financial losses and spectacular computer crashes, and have resulted in countless hours of debugging.
- Mature platform
  - Kotlin has been an officially supported language for building Android apps
- Concise, readable code
  - Language is designed to eliminate boilerplate code.
- Interoperable with Java/C#/JS/Python
  - Kotlin code compiles so that you can use Java and Kotlin code side-by-side, and continue to use your favourite Java libraries.

Benefits of Kotlin

Java Code

```java
public class Aquarium {

    private int mTemperature;

    public Aquarium() { }

    public int getTemperature() {
        return mTemperature;
    }

    public void setTemperature(int mTemperature) {
        this.mTemperature = mTemperature;
    }

    @Override
    public String toString() {
        return "Aquarium{" +
                "mTemperature=" + mTemperature +
                '}';
    }
}
```

Kotlin Code

```kotlin
data class Aquarium (var temperature: Int = 0)
```

Task: Install the Java Development Kit (JDK)

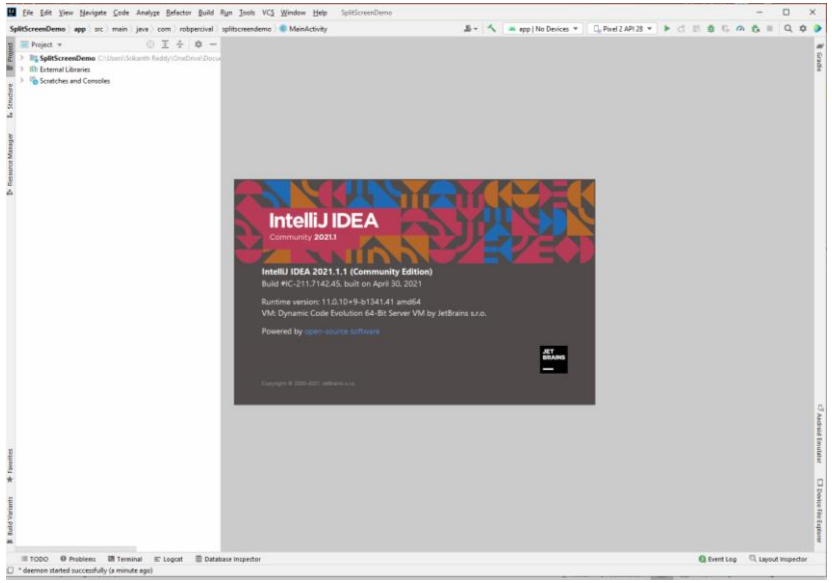
- You need to have the JDK installed to run Kotlin programs.
  - type **`javac -version`** in a terminal window.
- Step 1: Uninstall any older versions of the JDK/JRE
- Step 2: Download the JDK
  - You can download the JDK for free here: http://www.oracle.com/technetwork/java/javase/downloads/index.html
- Step 3: Install the JDK and JRE
  - By default, the JDK is installed in the `C:\Program Files\Java\jdk-14.0.1` directory

Task: Install the Java Development Kit (JDK)

- Step 4: Add the JDK installation directory to PATH (Windows only)
  - In **Settings** for Windows, search for **edit environment** in Find a setting.
  - Select **Edit environment variables** for your account in the list of matches.
  - In the **Environment Variables** dialog in the **User variables** section, select **Path** and click the **Edit**... button.
  - Add the path to the JDK's bin directory, for example, `C:\Program Files\Java\jdk-14.0.1\bin`, after any existing items.
- Step 5: Verify the JDK installation
  - To verify that the JDK was installed correctly, type the following commands in a terminal window:
    - `java -version`
    - `javac -version`

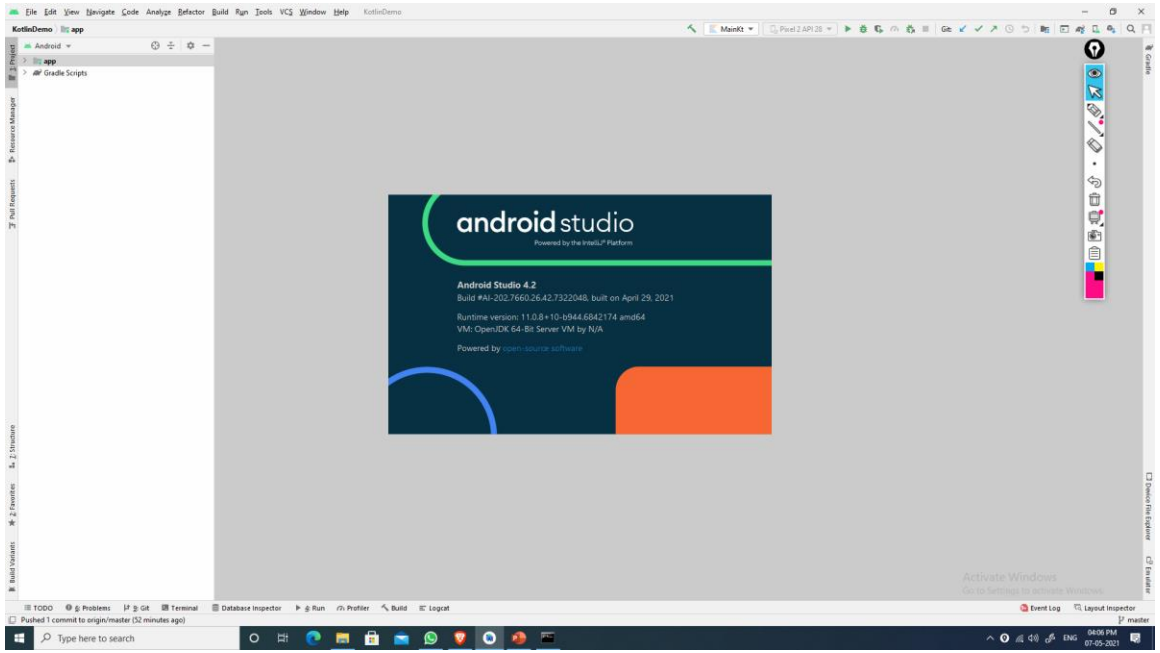
Task: Install IntelliJ IDEA

- Step 1: Download and install IntelliJ IDEA
  - Link : Download IntelliJ IDEA for your operating system.
- Step 2: Verify your IntelliJ IDEA installation
  1. Start IntelliJ IDEA.
  2. Install any updates and additional content you are prompted for.
  3. Select **Configure** > **Check for Updates** until there are no more updates available.



Task: Install Android Studio

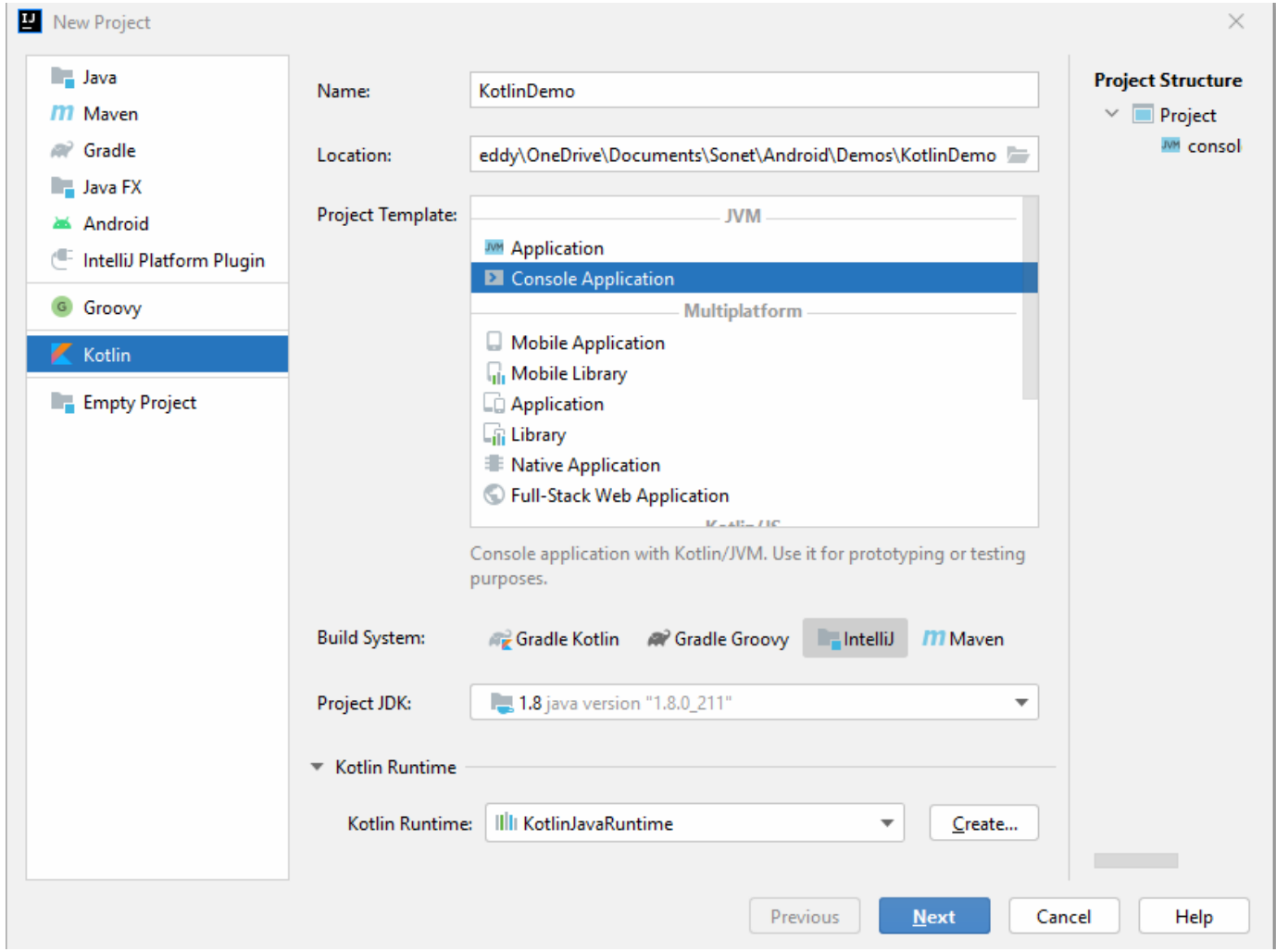
- Step 1: Download and install IntelliJ IDEA
  - Link : Download Android Studio for your operating system.
- Step 2: Verify your Android Studio installation
  1. Start Android Studio.
  2. Install any updates and additional content you are prompted for.
  3. Select **Configure** > **Check for Updates** until there are no more updates available.



Task: Create Hello Kotlin in IntelliJ IDE

- Create a Kotlin project so IntelliJ IDEA knows you're working in Kotlin.

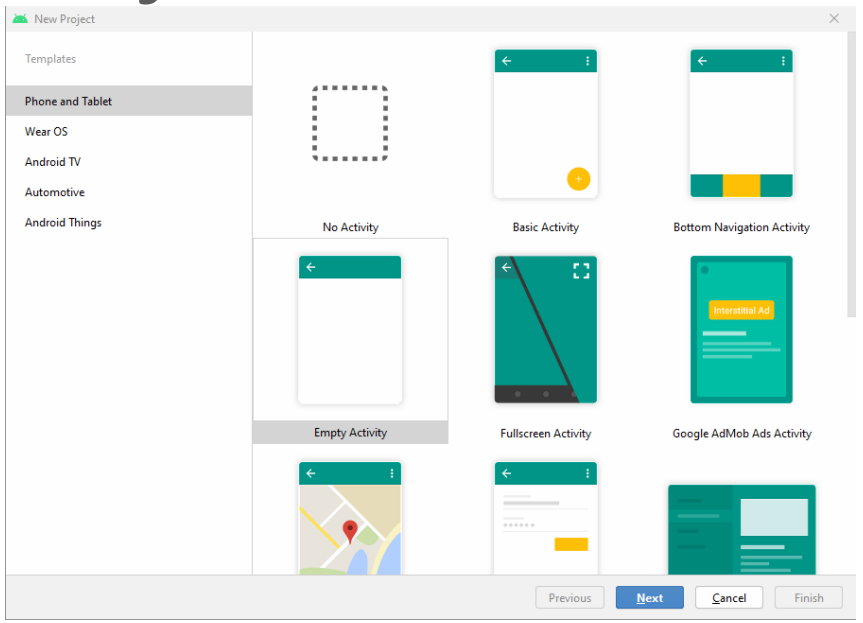
1. In the **Welcome to IntelliJ IDEA** window, click **Create New Project.**
2. In the **New Project** pane, select **Kotlin** in the left-hand navigation.
3. Select **Console Application**, JVM in the right panel and click **Next**.

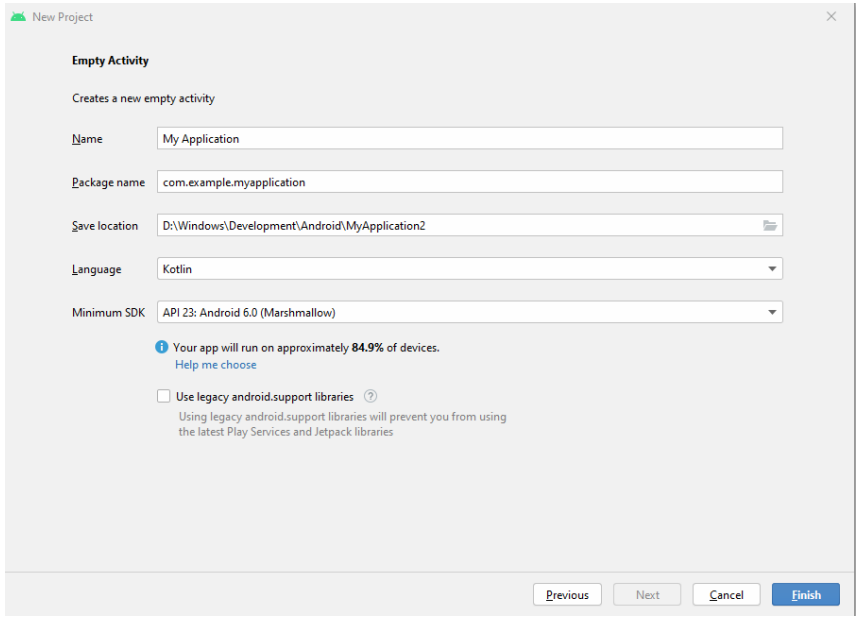


Task: Create Kotlin Mobile Project

- Create a Mobile project so Android Studio knows you're working in Kotlin.

1. In the **Welcome to Android Studio** window, click **Create New Project.**
2. In the **New Project** pane, select **Phone/Tablet** in the left-hand navigation.
3. Select **Empty Activity**, in the right panel and click **Next**.
4. Select **Kotlin** as language and **Minimum SDK** for mobile dev.





Task: Create Hello Kotlin