Program No:	**2**
Roll No :	**1545**
Title of Program :	**Android Project Files**
Objective :	**Why is a manifest file required in Android?2** **What is the activity main XML file?** **What is a Java file used for?**

1. Why is a Manifest File Required in Android?

The **AndroidManifest.xml** file is a critical component of every Android application. It serves as a central configuration file that provides essential information about the app to the Android system. Its key purposes include:

- **Declaring Components:** Registers all major app components, such as activities, services, broadcast receivers, and content providers.
- **Permissions:** Lists the permissions the app needs (e.g., internet access, camera usage), which are then requested from the user.
- **App Metadata:** Specifies general app information—like the app's name, icon, package name, and themes.
- **Intents and Filters:** Defines which components can respond to certain intents (e.g., opening a specific file type or link).
- **Minimum Android Version:** States the minimum (and sometimes target) Android API level the app supports.
- **Hardware/Software Features:** Declares required hardware (like GPS) and software features.

Without the manifest file, the Android system wouldn't know how to launch the app or what capabilities it needs, making the app unusable.

2. What is the Activity Main XML File?

The **activity_main.xml** file is typically the default layout file for the main activity in an Android application. It defines the **user interface (UI)** for that screen using XML markup, separate from the Java or Kotlin code. The file:

- **Describes UI Elements:** Specifies the structure and properties of UI elements (such as buttons, text views, images, etc.).
- **Layout Arrangement:** Arranges the UI elements using layouts (e.g., LinearLayout, RelativeLayout, ConstraintLayout).
- **Resource Reference:** Allows referencing UI IDs in code so that they can be manipulated by the activity's logic.

This separation between UI definition (XML) and code (Java/Kotlin) enables designers and developers to work more efficiently and maintain a clear structure in the app.

3. What is a Java File Used For?

In Android development, a **Java file** (typically with a `.java` extension) is used to implement the logic, behavior, and functionality of app components. Java files:

- **Define Classes and Methods:** Contain classes such as activities (`MainActivity.java`), services, or utility classes.
- **Handle User Interactions:** Manage event handling (e.g., button clicks) and connect to the UI via code, using methods like `findViewById`.
- **Business Logic:** Process data, control navigation between screens, and coordinate workflows inside the app.
- **Interact With Other Components:** Communicate with system services, databases, network resources, and more.

In essence, Java files bring to life the app's behavior and link the UI layouts with functional code that responds to user actions and system events.