University of Babylon – College of Information Technology
Information Network Department-Third Class

Mobile Fundamentals and Programming

Efficiency of Java Code for Mobile Application Development

Dr. Raaid Alubady - *Lecture 5*

1. Introduction

Mobiles applications are written in many programing codes. Mobile application beginners assume that Java programming best practices are equally applicable to mobiles applications programming. Android is one of the most popular mobile platforms paving the way for the development of a flood of mobile applications. Android applications are written in Java though there is some difference between the Java API and the Android API. So, many programmers think that mobiles applications development is easy for Java programmers. They also consider it’s important to understand and learn Android because of its dominance in mobile computing. In fact, a significant number of Java programmers start mobiles applications programming right away after reading a few tutorials.

2. Popular Coding Language

There are several programming languages to choose from. One of the easiest ways is to select the most popular language. According to Tiobe (see Figure 1), Java has been the number 1 or 2 and most popular coding language in 2017. Basically, since its creation in the mid-90’s. Many of the world’s biggest companies use Java to build desktop programming, mobile applications and backend web systems.

There are a number of factors that make Java so popular:

- ***Portability*** – because of Java Virtual Machine (JVM), Java can run on nearly every system. Java is also the most popular Android language, so the vast majority of Android applications are built in Java.

- ***Scalability*** – Java is built for scalability in mind, which is why it is so popular among enterprises and scaling startups. *Since Java is a statically-typed language*, it is faster and easier to maintain with less bugs. It is also backwards compatible, which means old versions of the language will still run perfectly even after new versions are released. This is a big relief for businesses who would otherwise worry about rewriting their code every time a new version comes out.
- ***Large community*** – The popularity of Java helps to ensure its future popularity, thanks to a huge community of users. With massive Stack Overflow and GitHub communities, developers can find help on virtually any problem they might encounter.

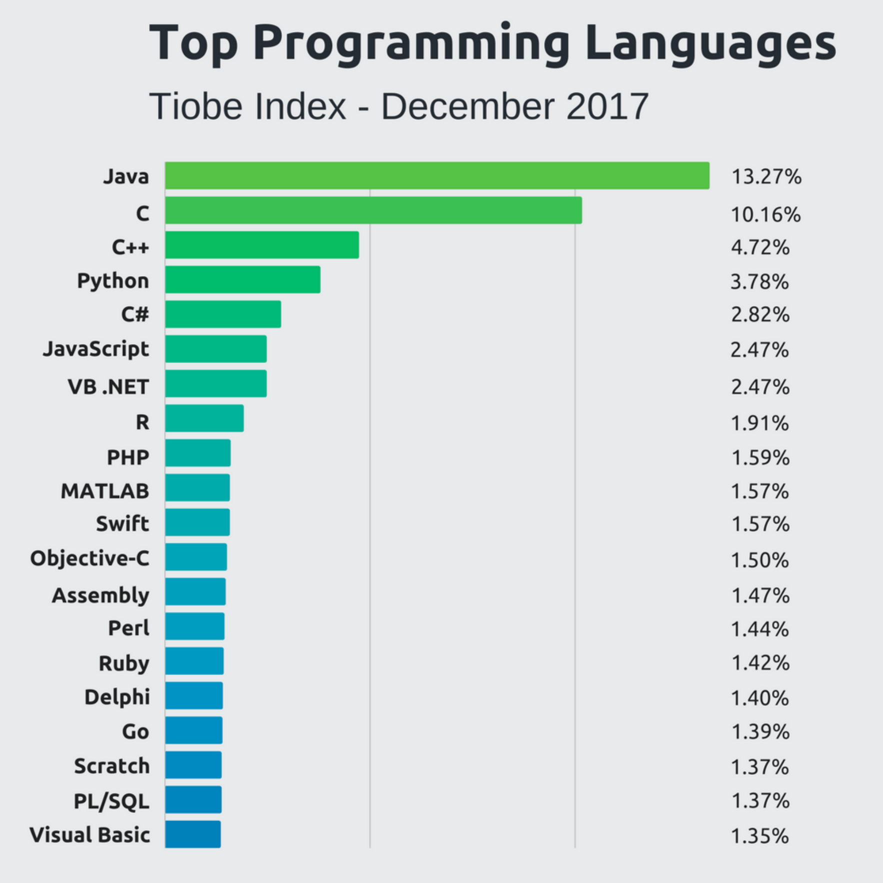

Figure 1. Top Programing Language

3. Types of Programming Languages for Mobile Application Development

Mobile application development industry in the last five years has multiplied in leaps and bounds, changing the way businesses function worldwide. With enterprises aligning mobile applications to their productivity in recent times, and with the rapid innovation in mobile devices across platforms, it calls for mobile application developers to write several versions of an application for many different platforms using a single language and many pieces of reusable code. We'll take a deeper look at five of the top languages using to develop mobile applications.

❖ C#

The most coveted programming language for *Windows Phone* application development, C# does the trick for *Microsoft* that Objective-C does for Apple. Although a Windows Phone platform couldn't emerge as the game-changer in the mobile application development industry, for loyal Microsoft users, C# makes the perfect programming language to build the robust Windows Phone applications.

Key features:

- Easy-to-use and a simple language
- Used extensively in the development of web applications and large tools
- It is a type-safe language
- It is scalable
- Access to .NET framework

❖ C++ Language

This is the most appropriate and robust programming language when it comes to building mobile applications for *Android and Windows*- and, mainly for low-level programming, it's still the go-to language on platforms for mobile application developers. As a powerful programming language, *C++ allows mobile applications to be developed for practically every purpose on every platform that exists.* It might not be super chic or trendy, but it dominated the programming world even before the smartphone revolution.

Key features

- C++ is one of the most powerful languages out there with tons of features.
- It is simple and efficient
- Object oriented
- Massive library
- Portable
- Extremely fast

❖ HTML5

HTML5 is the ideal programming language if you are looking to build a Web-fronted application for mobile devices. Although it makes various data types simple to insert, accounts for different screen sizes, rationalizes input parameters, and even levels the browser playing field, the problem with HTML5 is that it is still a proposed standard. Currently supported in a lot of different ways by a lot of different browsers, HTML5, from the cost-efficiency point of view, *has the advantage of building on the current version of HTML- making the learning curve much shallower than that for a completely new language.*

Key features:

- Easy-to-learn
- Support for mobile
- Responsive design with support for all the devices
- Drag-and-drop feature

❖ Swift

Swift is the latest programming language to foray into the *Apple* ecosystem, mainly considering its prevalence in writing code for Apple's latest APIs, Cocoa and Cocoa Touch. Even though it is a language written to work *along with Objective-C*, the Cupertino company is making it obvious for *iOS developers* to turn to Swift for complete programming. Designed to eliminate the likelihood of many of the security vulnerabilities possible with Objective-C, it's time for mobile application developers to turn to Swift, as many businesses are looking to hire Swift developers with expertise in developing cutting-edge mobile applications using this language.

Key features:

- Extremely easy to learn especially if you know Objective-C.
- It is open source.
- It is a simplified version of Objective-C.
- Easy-to-code.
- Maintenance is super-easy.
- It is the future of iOS development.
- It needs less coding as compared to other languages.

❖ Java Language

Java programming language is one of the most preferred languages when it comes to *Android application development*. An object-oriented programming language developed at *Sun Microsystems* (now owned by Oracle), Java can be run in two different ways: either in a browser window, or in a virtual machine that can do without a browser. This flexibility tends to mean a lot when it comes to re-using code and updating software. *Although Java does not have much to do if you are considering iOS development*, it certainly can be on your chosen list when it comes to mobile applications across platforms.

Java is used in development of:

- Android applications
- Server applications
- Web applications
- Embedded space
- Big data technology
- Scientific applications
- Websites
- Games

Key features:

- Object oriented language.
- It runs on all the platforms.
- Supports APIs that make integration a piece of cake.
- It is easy-to-learn and read.
- Hundreds of open source libraries available.
- Easy to get expert help from Android communities.
- Powerful IDEs make coding easy and error-free.

4. Java Programing and Android

When it comes time to develop Android applications, the first and most popular option is Java. Java is the official language of Android development, meaning it is the one that has the most support *from Google* and the one that most applications on the Play Store are built with. The number one way to develop Android applications, is to go ahead and download *Android Studio*. This is a piece of software called an *Integrated Development Environment (IDE)*. It will come packaged with the *Android Software Development Kit (SDK)* (i.e. a set of tools to facilitate Android development specifically) and basically this will give you everything you need in one place to get up and running.

- **Software Development Kit (SDK)**

Android development starts with the Android SDK. While there are many different programming languages and a host of IDEs can use to create an application, the SDK is a constant. SDK provides a selection of tools required to build Android applications or to ensure the process goes as smoothly as possible. Whether you end up creating an application with Java, Kotlin or C#, you need the SDK to get it to run on an Android device and access unique features of the OS. You'll also be able to use an ***emulator*** *to test applications you've built, monitor your device, and do a host of other things*. These days, the *Android SDK also comes bundled with Android Studio*, the integrated development environment where the work gets done and many of the tools are now best accessed or managed. You can download the SDK independently however if that is your preference.

- **Android Studio**

There are multiple ways to approach Android Development but by far the most official and powerful is to use *Android Studio*. This is the official IDE for the Android platform, developed by Google and used to make the majority of the applications that you probably use on a daily basis. *Android Studio was first announced at a Google I/O conference in 2013 and was released to the general public in 2014 after various beta versions.* Prior to its release, Android development was handled predominantly through *Eclipse IDE*, which is a more generic Java IDE that also supports numerous other programming languages. *Android Studio makes life significantly easier compared with non-specialist software*, but *is still has a little way to go before it can claim to be a completely intuitive and smooth experience*.

Good Luck

Regards