

Java Basics

► Programming Android Applications using Java!



Java Basics

- ▶ I don't want to spend a lot of time on teaching the basics of Java as we will be focusing more on the Android infrastructure.
- ▶ Very similar to C#
- ▶ Feel free to ask any questions at any time if what I am doing seems a bit strange!

Java Basics

▶ Variable scope

- ▶ Established using private/public/protected or absence of visibility modifiers for class-level variables
- ▶ Block-level scope for all other variables
- ▶ Primitive data types
- ▶ Reference data types (immutability of String)

▶ Classes

- ▶ Classes are templates for creating instances, or objects
- ▶ Can be viewed as a means for creating a specific types
- ▶ Can also contain static variables and/or methods where this makes sense

Java Basics

▶ Packages

- ▶ Used to group related classes and avoid naming clashes.
- ▶ Use the import statement to tell the compiler where to find classes.
- ▶ No import needed for classes in *java.lang* package.
- ▶ Analogous to namespaces in other MS languages.

▶ Varargs

- ▶ Variable length arguments in a method declaration
- ▶ Must be last parameter in the declaration
- ▶ Elements passed in can be tuples or an array
- ▶ Argument received are in an array

Java Basics

► Inheritance

- Used between classes where there is a strong *is-a* relationship.
- Java supports only single inheritance. A class always has at most one direct super class.
- All variables/methods that are not declared as private are inherited from the super class into the sub class.
- Sub classes can override and/or overload methods from the super class.
- Implemented by using the *extends* keyword.
- Look at Activity API documentation

Java Basics

▶ Interfaces

- ▶ A class-like construct that is used to define behavior
- ▶ Can only have abstract methods and constants
- ▶ Primarily used as contractual obligations in event-driven programming where an interface defines the signature of callback methods that a listener must implement
- ▶ We may implement our own when we talk about Fragment/Activity inter-communications
- ▶ Declared using the *interface* keyword
- ▶ Implemented using the *implements* keyword
- ▶ A class may implement more than one interface

Java Basics

▶ Callback methods

- ▶ Used extensively in Android
- ▶ For our purposes, methods that are created by us that are called by the runtime
- ▶ These are methods that typically won't be invoked explicitly by your code, but will run when specific events occur.

Examples:

- ▶ `onCreate()`, `onResume()`, `onStop()`, etc. are all methods that are invoked by the runtime due to changes in Activity lifecycle states
- ▶ `onClick()` is invoked by the runtime as a result of the user clicking a button.