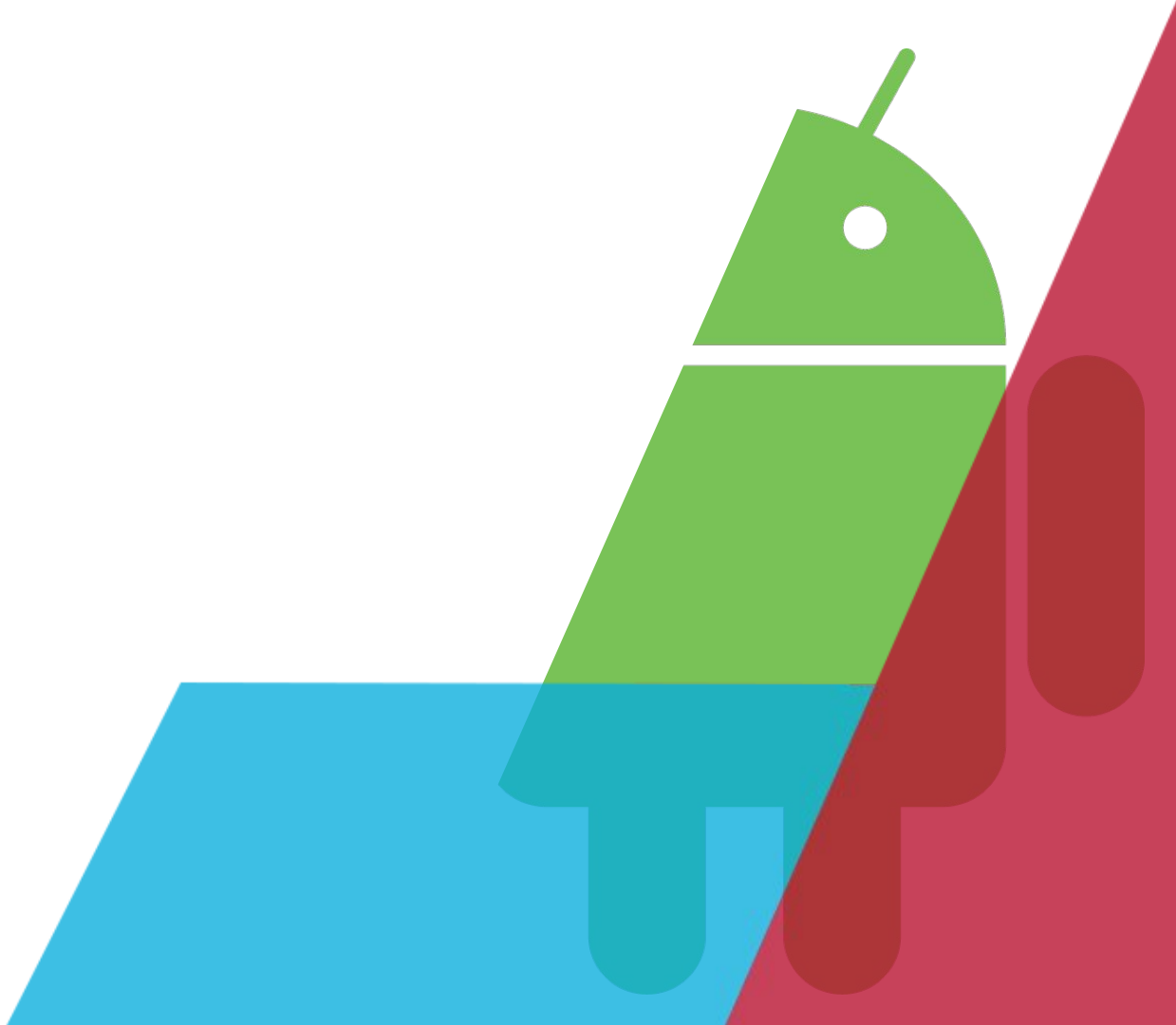
Functions



Upon completion of this module, a student will be able to

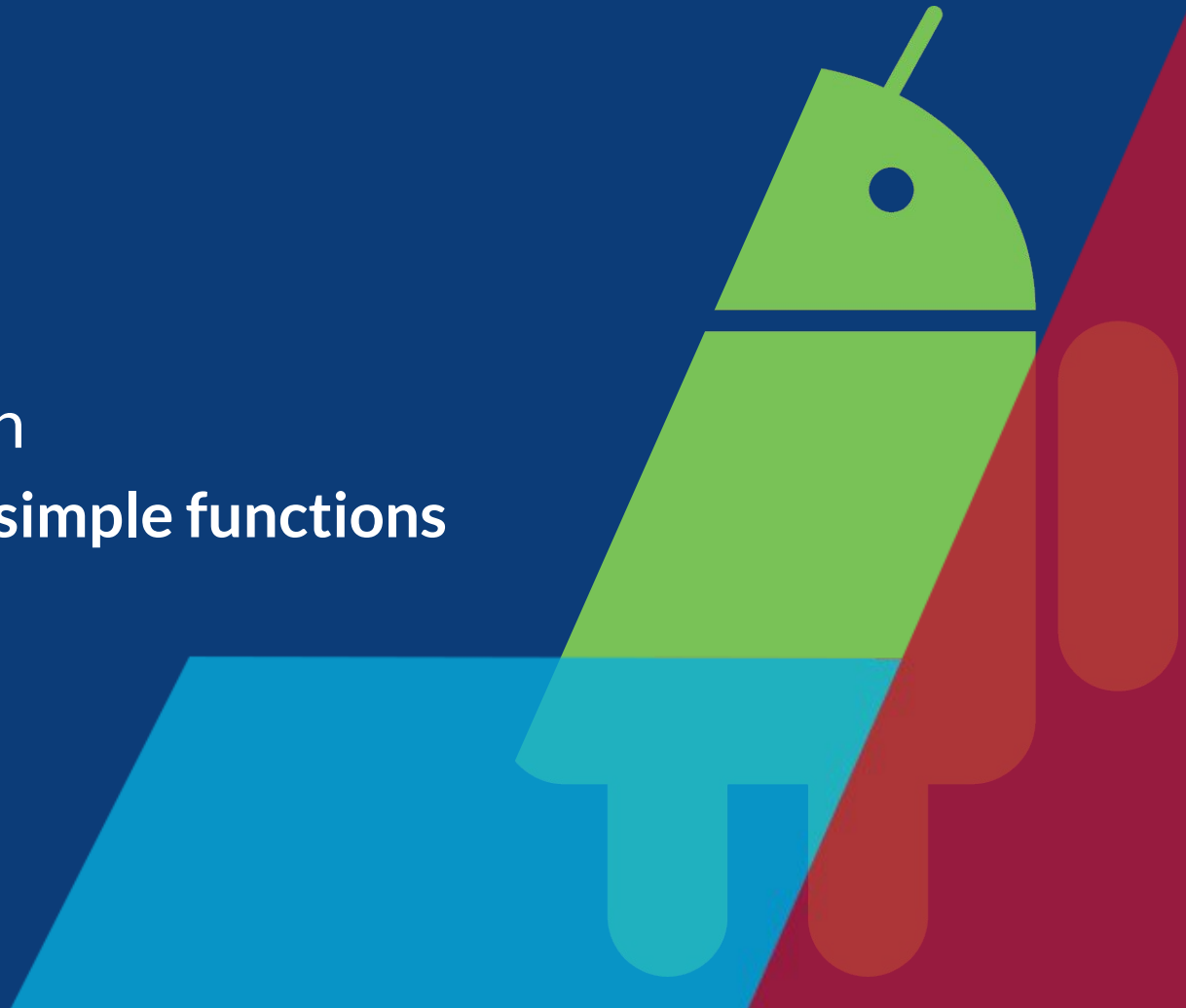
- write and call simple functions
- return a value from a function
- write functions that accept and use parameters
- work with named and default parameters
- overload functions so that they be declared with many parameter combinations
- accept text input from the user

Project

- Task
  - Write a guessing game where the user has to guess a secret number you will hard code as another data member. After every guess the program tells the user whether their number was too large or too small.
  - Must include these functions
    - int checkGuess(int guess)
    - void updateUI(int result)
- Repo
  - https://github.com/LambdaSchool/Android_Methods
- Challenge
  - Randomly generate a secret number.
    - https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.random/-random/index.html
    - When the user has correctly guessed the number, allow them to reset and try again.
  - Experiment with TextView attributes and other GUI components to improve the look of your app

A Student Can

write and call simple functions



functions

- sections of code that can be run from different places
- allows for modularization
- can return a result

Simple functions

- fun keyword
- Name
- Body
- function Call

```
performTask()

fun performTask() {
    // code for task
}
```

Challenge

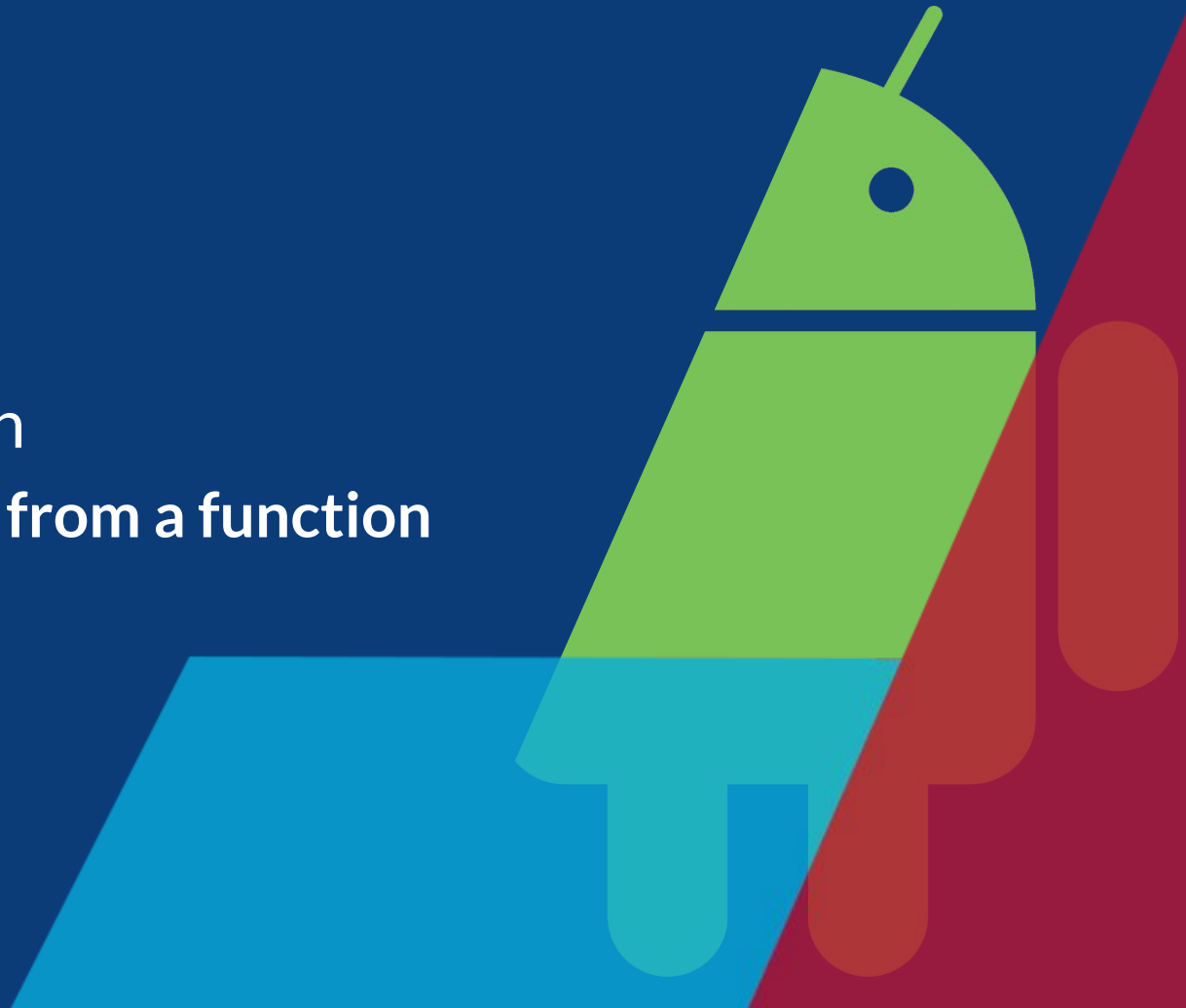
- Write a function

Solution

```
<LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MainActivity">

    ...

    <TextView
            android:id="@+id/my_text_view"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="This is my value" />

</LinearLayout>
```

A Student Can

return a value from a function



Return Values

- Return type
- Return a value
- Use returned value

```
val result = calculateValue()

fun calculateValue(): Int {
    val intvalue = 5
    // code for task
    return intvalue
}
```

Challenge

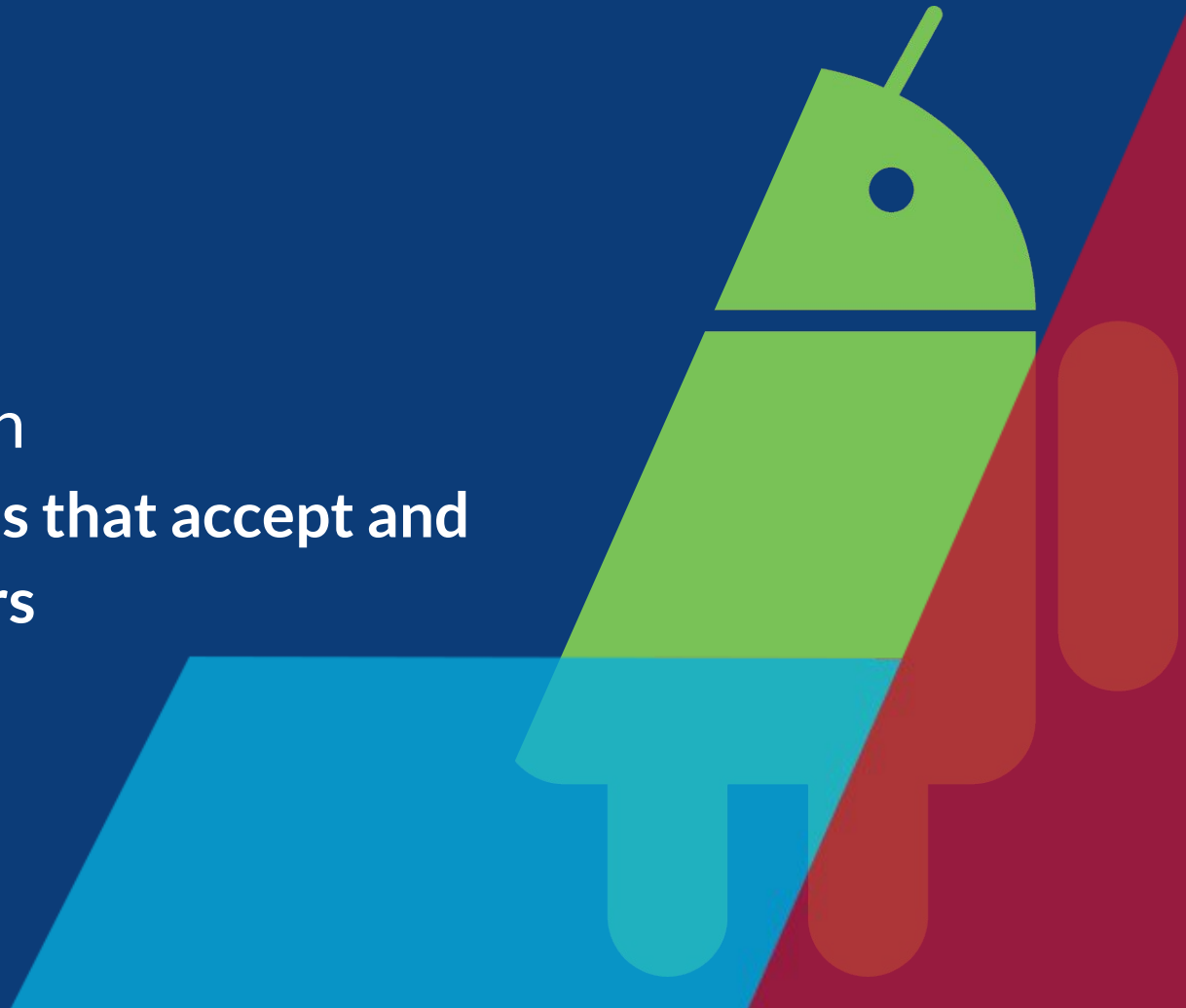
- Write a function to find and return the smallest number among two hard coded numbers.

Solution

```
fun findSmallest(): Int {
    val num1 = 5
    val num2 = 9

    return when {
        num1 < num2 -> num1
        num1 > num2 -> num2
        else -> num1
    }
}
```

A Student Can

write functions that accept and use parameters



Pass Parameters

- Parameters
- function Call
- Pass parameter

```
val result = calculateValue(12)

fun calculateValue(intvalue: Int): Int {
    // code for task
    return intvalue * 5
}
```

Challenge

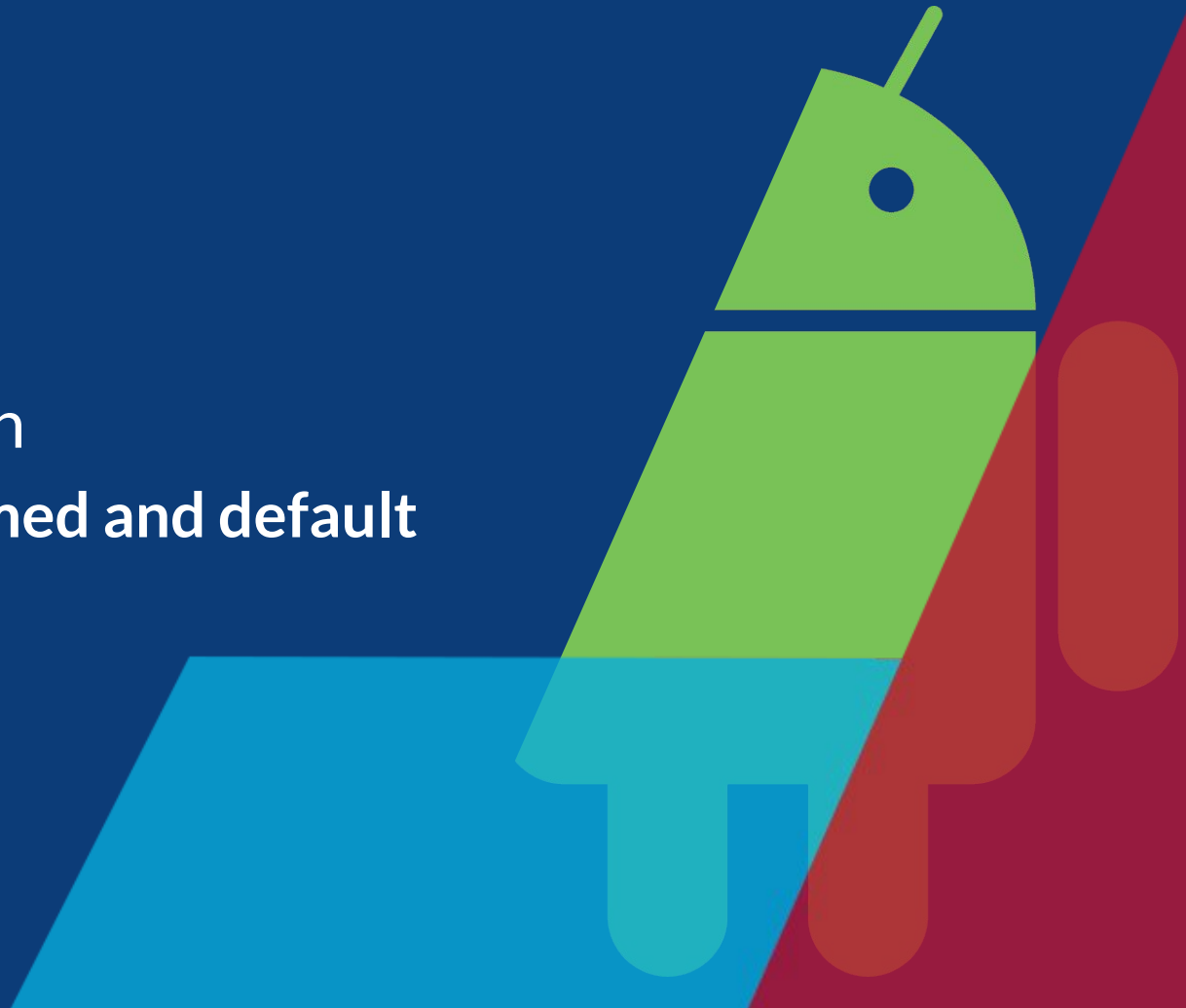
- Change your function to accept two parameters

Solution

```
fun findSmallest(num1: Int, num2: Int): Int {
    return when {
        num1 < num2 -> num1
        num1 > num2 -> num2
        else -> num1
    }
}
```

A Student Can

work with named and default parameters



Default Parameter Values

- Can provide a default if a parameter isn't passed

```
fun functionDriver() {
    println(calculateValue(4))
    println(calculateValue(4, 10))
}

fun calculateValue(multiplicand: Int,
                   multiplier: Int = 5): Int {
    return multiplicand * multiplier
}
```

Named Parameters

- Parameters can be passed by using their names
- Passed in any combination or permutation

```
fun functionDriver() {
    calculateValue())
    calculateValue(4))
    calculateValue(multiplier = 4,
                   multiplicand = 10))
    calculateValue(multiplier = 4))
    calculateValue(multiplicand = 10))
}

fun calculateValue(multiplicand: Int = 5,
                   multiplier: Int = 5): Int {
    return multiplicand * multiplier
}
```

Challenge

- Use default and named parameters to call your “findSmallest” function in a variety of ways

Solution

```
fun smallestDriver() {
    findSmallest(90)
    findSmallest(45, 90)
    findSmallest(num2 = 90)
    findSmallest(num1 = 45)
    findSmallest(num2 = 90, num1 = 45)
}

fun findSmallest(num1: Int = 0, num2: Int = 10): Int {
    return when {
        num1 < num2 -> num1
        num1 > num2 -> num2
        else -> num1
    }
}
```

A Student Can

overload functions so that they be declared with many parameter combinations

Overload functions

- Multiple functions with same name but different parameters

```
fun calculateValue(intvalue: Int): Int {
    return intvalue * 5
}

fun calculateValue(value: Double): Double {
    return value * 5
}

fun calculateValue(): Int {
    return 5
}
```

Challenge

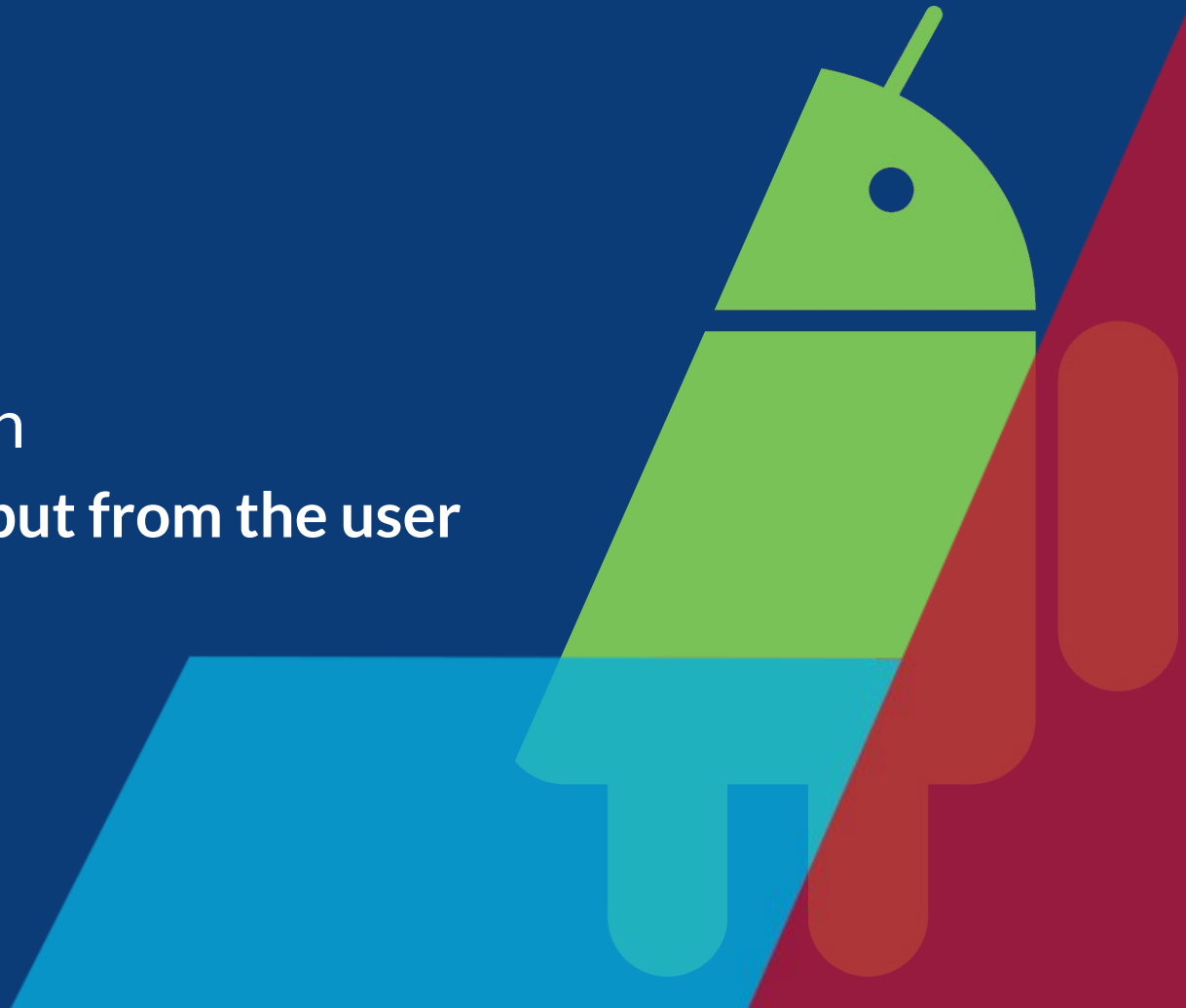
- Write a function to find the smallest number among three numbers, which can be passed in as Strings or Ints.
- Hint: use myString.toInt() to convert from a String to an int

Solution

```
fun findSmallest(num1: Int, num2: Int): Int {
    return when {
        num1 < num2 -> num1
        num1 > num2 -> num2
        else -> num1
    }
}

fun findSmallest(num1: String, num2: String): String {
    return when {
        num1.toInt() < num2.toInt() -> num1
        num1.toInt() > num2.toInt() -> num2
        else -> num1
    }
}
```

A Student Can

accept text input from the user



Without Graphical User Interface

- readLine()
  - Accepts input when user presses ENTER
  - Returns a string

```
println(inputDemo())

fun inputDemo(): String {
    return readLine() ?: ""
}
```

On Android

- EditText
- Text property
- toString

```
<EditText
   android:id="@+id/input_view"
   ... />
```

```
println(input_view.text.toString())
```

Challenge

- Write a function which reads a string from the user and converts it to uppercase using the “toUpperCase()” method and returns it
- *You’ll use the Android specific code for today’s assignment*

Solution

```
fun getUppercaseInput(): String {
    val input = readLine() ?: ""
    return input.toUpperCase()
}
```