

# Programming Fundamentals for Android

Utilize Data

January 21, 2016

# Corresponding Text

*Learn Java for Android Development*, pp. 1-11, 37-70

# IntelliJ IDEA Java IDE



- IntelliJ is a Java integrated development environment (IDE)
- Available at <https://www.jetbrains.com/idea/download>
- Requires the Java development kit (JDK) available at <https://www.oracle.com/technetwork/java/javase/downloads>

# Our First Java Program

```
package com.myname.week_01;

public class Main {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

We'll often use `System.out.println()` to display data at the console.

# Variables and Data Types

- A **variable** is a named location in the computer's memory use to store a value.
- A **data type** is a classification of data items and specifies possible values.

# Primitive Data Types

Data Type	Description
boolean	two possible values: true and false
char	a single Unicode character such as "A", "z", "5", or "%" with integer value between 0 and 65,535
byte	an integer value between -128 and 127, uses 8 bits of memory
short	an integer value between -32,768 and 32,767, uses 16 bits of memory
int	an integer value between -2,147,483,648 and 2,147,483,647, uses 32 bits of memory
long	an integer value between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807, uses 64 bits of memory
float	a 32-bit value used to represent numbers with fractional parts
double	a 64-bit value used to represent numbers with fractional parts

# Reference Data Types

- A variable with a **reference type** stores a reference, a memory address, to the location in memory that stores the value associated with the variable.
- Compare this to a primitive type which stores the value directly.
- An example of a reference type is the **String** type. A string is a sequence of characters.

# Creating Variables and Storing Values

- Variable names can be any collection of letters, digits, “\_”, or “\$” as long as the name doesn’t begin with a digit.
- Variables must be declared before they can be used. To declare a variable, we must specify both a type and a name.
  - `float temperature;`
  - `int zipCode;`
  - `boolean isRaining;`
  - `String cityName;`
- We can declare several variables of the same type at once.
  - `float temperature, humidity, pressure;`



# Arrays

- We can use a single variable to store multiple values of a certain type.
- **Array types** are reference types that allow us to store values in equal-sized, contiguous locations in memory.
- Individual values in an array are known as **elements**.
- Brackets, [ and ], are used in array declarations.
- Examples:
  - `int[] dailyHighs;`
  - `String cityName[];`
- Usually see brackets next to type except when declaring multiple variables.
  - `float currentTemperature, forecastHighs[], forecastLows[];`

# Expressions

- An **expression** is a combination of literals, variable names, method calls, and operators.
- A **literal** is an value expressed verbatim like the integer *123* or the string *"hello"*.
- There are several kinds of literals: a single character, a string, an integer, a floating-point value, a boolean value, and null (used to indicate that a reference value doesn't refer to an object).

Data Type	Literal Examples
char	'1', 'A', '?'
String	"Hello", "Goodbye"
int	1, -20, 1234
double	1, 2.0, 0.828, -123.92929
boolean	true, false

# Expressions

- A **simple expression** is an individual literal, variable name, or method call.
- A **compound expression** is a sequence of simple expressions and operators.
- An **operator** is a symbolic instruction used to transform data, the operands, into another value.

# Arithmetic Operators

Operator	Symbol	Description
Addition/String Concatenation	+	Returns the sum of two numeric operands or the concatenation of strings
Division	/	Returns the quotient of two numeric operands
Multiplication	*	Returns the product of two numeric operands
Subtraction	-	Returns the difference of two numeric operands
Unary Plus	+	Returns the numeric operand
Unary Minus	-	Returns the arithmetic negative of operand
Increment	++	Add one to the operand, prefix and postfix operator
Decrement	--	Subtract one from the operand, prefix and postfix operator

# Other Operators

Operator	Symbol	Description
Array Index	[ ]	Given <b>variable[index]</b> , where <i>index</i> is an integer, return or store the value in <i>variable's</i> storage at index
Assignment	=	Given <b>variable = operand</b> , store <i>operand</i> in <i>variable</i>
Compound Assignment	+=, -=, *=, /=	Given <b>variable operator operand</b> , perform the operation using the existing value of <i>variable</i> and <i>operand</i> , store the result in <i>variable</i>
Comparison	==, <, <=, >, >=	Given <b>operand1 operator operand2</b> , compare both operands, return <i>true</i> or <i>false</i> depending on the comparison; includes equality, less than, less than or equal to, greater than, greater than or equal to
Object Creation	new	Given <b>new identifier[integer size]</b> , allocate a one-dimensional array of values with type <i>identifier</i>

# Examples

# GitHub

- We'll be using GitHub to share our work and to submit assignments.
- Built on Git, a version control system used to track changes.
- Four distinct parts: the working directory, the staging area, the local repository, and a remote repository.
- For an overview of Git, see <https://try.github.io>.
- Sign up for a GitHub account at <https://github.com>.
- We can work with GitHub directly from IntelliJ.

# First Project

- Project 1 should make use of topics covered during weeks one through three.
- Create a program that allows a user to add, remove, edit, and list to-do items by presenting the user with a menu similar to the following:  
Please choose an option:  
(1) Add a task.  
(2) Remove a task.  
(3) Update a task.  
(4) List all tasks.  
(0) Exit.
- If the user chooses to add a task, the program should prompt for a description of the task and add it to the collection of tasks.
- If the user chooses to remove a task, the program should ask the user which item to remove and remove it from the collection of tasks.
- If the user chooses to update a task, the program should ask which task will be updated and for a new description of the task.
- The program should loop until the user chooses to exit.
- The program should include methods dedicated to adding, removing, and updating tasks.



# Exercise

Write a program that stores a city name as a string, the city's zip code as an integer, and the high temperature for the next five days in an array. The program should calculate the average high temperature for the next five days and store the value in a variable. Finally, the program should display the city name, the zip code, and the average high temperature.

The average of a collection of values is the sum of all the values divided by the number of values.

For example, if the city is Columbus; the zip code is 43215; and the high temperatures for the next five days are 32, 25, 27, 40, and 45, the output of the program might be:

City: Columbus

Zip Code: 43215

Average High Temperature 33.8

When computing the average high temperature, was the value stored as an integer value or a floating-point value? What happens to the computed average with either data type?

**Be sure to commit your code and push it to GitHub.**