

GENERAL ASSEMBLY - JAVA FOR ANDROID

02/01/2015

Download - <http://git-scm.com/download/>

- Note the version of operating system and version

GIT & GITHUB

- Version control history
- git config --global user.name "NeelRoshania"
- git config --global user.email "nroshania@gmail.com"

Create a GIT repository

- Staging files
 - Changes are made and ready to save them - add
 - Save all changes made - commit -m 'INSERT A MESSAGE'
- Process of adding a repository and
 - git init - create a local git repository
 - git add .
 - git commit -m "message"

GIT HUB

- Service that allows you to host git repositories
- Allows for easy sharing of code
- Public and private repository
 - Bit bucket

PUSH, FORKING & CLONING ON GITHUB

- Fork - Make a copy of a GITHUB's user repository on your profile
 - Use the fork button on GITHUB
- Clone - Make a copy of the fork onto a local repository (aka your computer)
 - Copy link in repository
 - git git clone *instructorsRepositoryLink*
- Make the desired changes.
- Save changed file on a separate local repository
- Specify folder location on terminal
 - cd desktop/localRepositoryPushFolder
- Push to changes to the GITHUB cloud
 - git push
 - git push -u origin master

FUNDAMENTAL OF JAVA

1. Install Java JDK

- a. <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

2. Install IntelliJ

- a. <http://www.jetbrains.com/idea/download/download-thanks.html?platform=mac&code=IIC>

JAVA

- Handles logic functions of app
- Fork, Clone and Pull request for share and edit data amongst other developers

https://tlk.io/java_workshop

INTRODUCTION

- Functions and uses
 - Portability of code across different applications
 - Used to develop Android applications
- History
 - Write once, use anywhere.
 - James Gosling
- Characteristics
 - Procedural vs Object oriented programming
 - Needs to be compiled
 - HTML requires no compiling
 - Compiler translates text into machine code
 - 1 and 0
 - IntelliJ
 - Write, compile and run java code any application of Java
 - Android Studio is specific to android applications

INTELLIJ

- **Create a new project**
 - Select Java
 - Define version of Java installed
 - New...JDK
 - Choose JDK root
 - Next
 - Choose project name
 - OK

•

- **Create a new class**

- Click "src" source folder
- Define new class

- **Structure**

- IntelliJ shortcuts
 - "psvm" - "public static void main" function
- Primitives and Objects
 - Primitives - Basic data types
 - Integers
 - Doubles and floats
 - Strings
 - Short, long, byte etc
 - Boolean - True/false
 - Objects
 - Collections of data structured in a certain way
 - <https://docs.oracle.com/javase/tutorial/java/concepts/>
 - "Identifying the state and behavior for real-world objects is a great way to begin thinking in terms of object-oriented programming."
 - You may also notice that some objects, in turn, will also contain other objects. These real-world observations all translate into the world of object-oriented programming.
 - Identify object
 - Identify number of states
 - State at a point in time
 - Identify behavior
 - Methods of changing between states
 - Software objects
 - Stores states in variables/fields
 - Stores

- **Declaration or assignments and initialization**

- Type variableName = Value;
 - int x = 12;
- String manipulators
 - "\n" - new line

- **Functions/methods**

- Defined within a class
 - Cant define methods outside of classes
 - Define whether the method assigns of executes
- Save development space
- Can make one change and effect a large set of numbers
- Characteristics
 - <https://docs.oracle.com/javase/tutorial/java/javaOO/methods.html>
 - Modifiers—such as public, private, and others you will learn about later.

- The return type—the data type of the value returned by the method, or **void** if the method does not return a value.
 - The method name—the rules for field names apply to method names as well, but the convention is a little different.
 - The parameter list in parenthesis—a comma-delimited list of input parameters, preceded by their data types, enclosed by parentheses, (). If there are no parameters, you must use empty parentheses.
 - An exception list—to be discussed later.
 - The method body, enclosed between braces—the method's code, including the declaration of local variables, goes here.
- **Process**
 - Define method
 - Call method
 - **Booleans, operators and conditionals**
 - Operators
 - <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html>
 - Conditionals
 - <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/op2.html>
 - IF ELSE
 - if (*condition*) {} else {}
 - Boolean
 - True or false
 - 1 or 0
 - isTraining = (3>2)
 - True as 3 is indeed greater than 2

USEFUL FUNCTIONS/OBJECTS

- Type *Scanner*
 - Read information from command line
 - Scanner (*Define input*)
 - System.in (input from command line)
 - Need to declare following to utilize function
 - Scanner input = **new** Scanner(System.**in**);
 - Request String - nextLine
 - Request integer - nextInt
- .equals(Variable type)
 - Determines if strings or integers are unique