1. Syllabus

2. Topics

Interviewing sucessfully

Node Sync/Async

Survey design

Android-Kotlin

Annotations

Basic Control Flow

Basic Syntax & Types

Classes

Constructor Default Values

Data Binding

Data Classes

Delegated Properties

Delegation

Enum Classes

Functions

Generics

Gradle

Grid Layout

In Line Functions

Interface

Internet Data and Images

Kotlin Introduction

Nested Classes

Null Safety

Properties and Fields

Returns and Jumps

Sealed Classes

Set Up

Work Manager

Angular Elements

Angular material

Angular testing with Cucumber and Protractor

Angular unit tests

Basic introduction to Redux

Big Picture of Web Dev

Clean Code

Clean Code for Data Science

Clean Code for Java

Clean Code For JavaScript

Clean Code for Kotlin

Clean Code for Python

Clean Code for SQL

Clean Code for XML

General Clean Code Guidelines

Curious learning and research strategy

MEDIUM: Data Ethics

Data Ethics and Privacy

Data Validation

Data validation and quality control

ElasticSearch

Express JS

Free Code Camp

FreeCodeCamp - Data Analysis with Python

FreeCodeCamp - Maths for Programmers - Sets and Logic

FreeCodeCamp - Scientific Computing with Python

JavaScript Algorithms and Data Structures

FreeCodeCamp - Basic Algorithm Scripting

FreeCodeCamp - Basic Data Structures

FreeCodeCamp - Basic Javascript

FreeCodeCamp - Debugging

FreeCodeCamp - ES6

FreeCodeCamp - Functional Programming

FreeCodeCamp - Intermediate Algorithm Scripting

FreeCodeCamp - Object Oriented Programming

FreeCodeCamp - Regular Expressions

Legacy Responsive Web Design

FreeCodeCamp - Applied Accessibility

and HTML5

FreeCodeCamp - CSS Flexbox

FreeCodeCamp - CSS Grid

FreeCodeCamp - Responsive Web Design Principles

Git Basics

Git and GitHub

Introduction to Git and GitHub Part: 1

Introduction to Git and GitHub Part: 2

Version Control & Scrum using Trello & Github

Git feature branching

Google forms like a boss

High performance dev teams

How to ask for help with your code

How to be a professional remotely

How to download youtube videos from the terminal

Intro to CSS architecture resources

Intro to functional programming

Intro to hybrid mobile

Intro to software architecture

Introduction to Ajax

Introduction to assertive programming

Introduction to Design Patterns

Introduction to Dotenv module

Introduction to Node and SQL

Introduction to pair programming

Introduction to Typescript

Introduction to unit tests with Mocha

Ionic

Introduction to Ionic.

Ionic Tools and Patterns

Progressive Web Apps.

iOS Mobile

Alerts Notifications and Application Life Cycle

Core Motion and Camera

More Segues

More Swift, Gestures and Animations

Multithreading, Auto-Layout and Other Functionality

Swift and More

Swift TDD and Unit Testing

Jasmine Spies

Jasmine unit tests

Java

Environmental variables and secrets with gradle

Exception Handling

Gradle and IntelliJ project submission structure

Intro to JUnit

Introduction to Gradle with IntelliJ

Introduction to Spring Boot

Intro to spring boot - Part 1

Intro to spring boot - Part 2

Intro to spring boot - Part 3

Spring Boot - Consume API

Introduction to Spring Infrastructure

Introduction to Spring Infrastructure part 1

Introduction to Spring Infrastructure part 2

Introduction to Spring Infrastructure part 3

Java collections and data structures

Java File IO

Java Generics

Java Lambda expressions and Functional Interface

Java learning materials

Java Logging with Log4j

Java Multithread, Concurrency and Parallelism

Java OOP basics resources and readings

Java project submission requirements

Java Streams

JDBC templates

Strings and numbers in Java

Javascript error handling best practices

JS and Node

APIs and Node

DOM events with vanilla JavaScript

DOM manipulation with vanilla JavaScript

ExpressJs

JavaScript template engines

Logging in Node and JS

MongoDB and JavaScript

Node

Node setup

Super basic intro to Node

The dotenv module

Using the fetch api

Jupyter notebooks best practices

Keep on growing!

Khan Academy

Khan Academy - Statistics and probability

Unit 1 - Introduction to Statistics

Unit 2 - Intermediate Statistics

Unit 3 - Advanced Statistics

Pre-Algebra

Seventh Grade Maths

Kinds of web APIs

linux

Environmental Variables

Introduction to Bash and the terminal

Introduction to Linux

Logging

Mocking with Angular

Model Evaluation

Nodejs intro to socketIO

optimizations and big-O

Pull Requests

python specific resources

Automated Testing in Python

Intro OOP(Object-oriented Programming) for Python

Introduction to APIs

Logging in Python

MongoDB and Python

Python and Kafka

Python and Spark

Python backend dev environment setup

Python Decorators - Advanced

Python Decorators - Intro

Python Exception handling

Python Iterators and Generators

Python Lists in depth

Python Recursion by example

Python writing custom exceptions

SQL and Python

Sqlalchemy ORM

Sqlalchemy Basics

Sqlalchemy migrations with Alembic

Virtual Environments

Web scraping with Python

Python versus JavaScript

React and Redux Developer tools

Redux

Redux Saga

Redux Thunks

Regular Expression Resources

Reopening a Pull request

strategy department topics

facebook marketing

google analytics

Google Sheets

mailchimp

SQL for strat

statistics for strat

Test Driven Development

Test Driven Development - Questions

The Tech Landscape Terminology

Unit testing (language agnostic concepts)

TDD Horrors

Unit testing with mocks and spies

Web Dev learning materials

Web Frontend

Intro to Angular

Intro to Vue

Introduction to CSS

Introduction to web design

React

Intro to React

Official React Tutorial

React + Redux architectural guidelines

React Unit testing

React: Basic basic architectural guidelines

Storage Options

What to put in your CV

3. Workshops

4. Projects

\_depricateds

Agiles

Coding\_aptitude\_assessments

Common problems

Courseras

Credits

Data-sciences

Design-thinkings

Devops

Docker

Employability-sprints

Employed-skills-tests

File-and-directory-namings

Getting set up to write code on your device

Gradles

Javascript debugging in VSCode

Kubernetes

Language agnostic

Learning how to teach coders

Medium-touch-course-commons

National Qualifications Framework

Onboardings

Professionalisms

Project-submission-instructions

React-natives

Scrums

Soft-skills

Specific skill assessment criteria

SQL

Teaching-curriculums

Tech-big-pictures

Why and how to review code

Zmc-challenges

More

 Github repo

 Credits

 Download

 Star26

 Fork82

TOPIC: INTRODUCTION TO TILDE

Tagsskill/tilde

Welcome to Tilde. We built it with love ❤️

If you are new to agile project management, it’s going to look a little weird at first. Don’t worry. Before you get

too deep into how we actually use this thing, it’ll be good for you to know a bit about the principles and practices that inspired the design.

Please Note: One of the login mechanisms uses a popups, if you are using ad blockers and popup blockers you will need to turn them off. Please also ensure cookies are enabled in your browser.

Here is little video tour of the first features you’ll be interacting with, please note there are 3 videos in this playlist.

These videos don’t cover everything. Please read this whole article, there is more text under the video.

How good agile teams work

The first thing to know is that working on a team is hard. And most techies and professionals work on teams. Back in the early days of software development things sucked. A lot. Software projects generally just failed; even big ones backed by people with really solid technical skills. Projects were mostly either late, broken, over-budget, or the features were just wrong. If you want success in this line of work, knowing how to use tech and any tools of your trade is just the tip of the iceberg lettuce.

So, after lots of suffering, the software industry started figuring out ways of working that are actually effective. There are loads and loads of things to be said about those methods. But the main thing to know for now is that we designed Tilde in such a way as to get you to practice

those methods. The way you will be working through this course is really quite similar to how industry giants build epic software.

It is also important to note that these principles and practices have been adopted over many industries, and agile project management techniques are still being adopted and adapted for other industries.

Kanban

Basically, Tilde is based on a thing called Kanban. Read this excellent article to understand what Kanban is all about: https://kanbanize.com/kanban-resources/getting-started/what-is-kanban

Ok cool! So, Tilde is basically a Kanban

board. Each card you see on the board is a piece of work that you need to get done. And it is your mission to get all the cards from the Backlog to Complete.

It’s not just about moving cards

And of course, the whole point of this thing is to make sure you get the skills you need to launch your successful career. This is not a box-ticking exercise. In getting your cards to Complete you need to make sure you really understand all the concepts that the card is about. Don’t race to finish, the point is not to move as quickly as possible. The point is to understand.

So please ask questions when you need to :)

And learn to ask questions well! (Yes, this

is a skill you can develop)

One major difference between a good coder and a bad coder is: Good coders UNDERSTAND the code

The goal of any course is to gain skills and understanding. Rushing through content will not help you in the long run. In fact, it will hinder you overall and could even damage your career. It is important to get into good habits around always optimising for understanding now.

Check your understanding!

It’s important to understand everything before you move forward. Test yourself with these quiz questions:

What is Tilde and why was it created?

What should you know before using Tilde, according to the text?

What is the purpose of the Kanban board in Tilde?

Why is it important to not rush through the content of the course?

What is the main difference between a good coder and a bad coder?

RAW CONTENT URL