Summer Prepwork Guidebook

Contents

Welcome to OC Tanner and the Summer of Learning

We are very excited to begin learning with you. Over the next 12 weeks you will be learning how to think like a developer, how to write programs in both Java and Angular.js, and how developers work *in the wild*.

The Summer of Learning is open to anyone who wants to learn with us. We will host weekly meetings to give participants a place to discuss what they learned during the week, as well as introduce some topics not covered in the self-paced courses. We will periodically have project days so that we use our new knowledge creatively and work together. Those who will be interns in the Fall (or are on the alternates list) are strongly encouraged to come to the weekly meetings, but they are not required.

We hope that you find this collection of resources helpful and entertaining. If you have any questions or feedback for us, please share with us as soon as you can. This will allow us to help you while it is relevant, and if an issue may affect others, we can get information to them quickly.

Once again, welcome! We hope you are as excited to get started as we are!

Calendar for Summer study groups

Discussion/ speaker days will allow time for questions to be answered/ pair work with mentors as well as a group discussion about a cool topic.

Date	Plan	Topic
June 4	Discussion	**Welcome!** & open topics
June 11	Project (2 in 1 day!)	**Design project:** Tearing down and mocking up a client's website
…	…	**Software Engineering project:** Making user stories, project management and pseudo-coding
June 18	Discussion	**Get ready to learn Java!** Environment setup, learning strategies
June 25	Discussion	**Apprenticeship Patterns** How good developers teach themselves.
July 2	No meeting for holiday	Maybe have a study group this day? Come with questions, or we can do some exercises together.
July 9	Discussion	**Design Patterns** Thinking about software design and architecture.
July 16	Discussion	TBD
July 23	Project	**Make a game in Java**
July 30	Discussion	TBD
August 6	Project	**Round-robin Java problems**
August 13	Discussion	TBD
August 20	Discussion	TBD
August 27	Project	**Angular app- group input to decide topic**

Knowledge Expectations

When we begin the internship in the Fall, our goal is to have everyone on the same base level. This will aid in pairing, so that when you pair, the mentor will know where your knowledge is and can set goals and expectations from there.

If you follow this guide and the provided calendar, you will have met the expectations we have for interns entering this Fall.

If you begin to feel like you are behind, or having to leave things 'half-finished' to keep up with the calendar, reach out to a mentor or Amanda early and often! Treat your learning the way you would an Agile project. Self-evaluate and have conversations with others early and often to avoid large problems down the road!

These are the technologies you need to be familiar with.

- HTML/ CSS
- JS
- Angular.js
- git & GitHub
- Java
- Computer Science concepts
- Software & Web development cycle

Intro to Software Design & Process

Learning Outcomes:

The Design Process

- Describe the design process
- How does the designer involve the client/user in the design process?
- What are some ways of communicating with a user to understand their needs & goals?
- What is friction, as it applies to design?
- What tools exist to help the designer reduce friction?

On Visual Design

- What is meant by "good design?"
- What are some established design patterns for visual content?
- What information is each pattern best suited for?

- How does the grid system help designers design sites?
- What advantages/ disadvantages do you see in this method?
- How can you use typography to influence a visitor's actions in a web page?

On Mockups

- What is a mockup? How are they used in design?
- Think about the tradeoff between time spent planning design vs. designing as you go. Consider the client reactions to these methods as well.
- Use mockups to plan different kinds of web sites. (commerce, personal, large company, small business, tools/ apps)

On the Principles of Software Engineering

- How do software engineers approach a problem? How do you think about problems? Did this section influence your approach to problem solving?
- Is software engineering only concerned with writing code (a product)? Why would process be so important?

On Agile Development

- What were some shortcomings of the "old" project management techniques?
- How does Agile development address these problems?
- Describe XP and some techniques employed by this method
- What is SCRUM?
- How do we get a user story from a product?
- Describe what a user story contains? What should a user story not contain?
- Describe the steps to take in completing an Agile project. (Perhaps by starting with "Gathering requirements…)
- Describe Test Driven Development (TDD) and Behavior Driven Development (BDD)

On Pseudocoding and Good Software Design

- Why do we use pseudocode?
- What is modularity?
- Pseudocode a common task, like doing laundry.
- How does a modular approach to software design help improve design?
- What are the SOLID principles of Object-Oriented software design?

Introductory Computer Science

Learning Outcomes:

Unit 1

- Summarize the history of computing
- Can you say a few sentences that describe fundamental hardware and software concepts
- Explain how the JVM translates Java code into executable code
- Explain Object-Oriented Programming concepts such as objects, classes, inheritance and polymorphism
- Write simple programs using basic Java concepts
- Describe and use primitive data types in Java
- Describe and use logical and relational operators and compare Boolean expressions
- Explain and use various control structures such as methods, decision statements, and loops
- Use an exception handling mechanism
- Use Java input/output class hierarchy to read and write data to and from external files.

Prepwork for Fall 2015 Internship

I. Software design and foundations

Viking Code School Prep

http://www.vikingcodeschool.com/prep

Timeline	**–**
Time estimate to complete	**10 days @ 2-3hrs/day**
Begin	**June 1**
Project day	**June 11**
Complete by	**June 12**

No sign-up required. This is a free tutorial.

Why this course?

The Viking School is an online web development program that has put together this prep series primarily to assist those interested in applying for their program. It is free for all and is a great intro to what web

development is all about. You are not required to complete the whole prep course, but we encourage you to explore the modules and learn any parts that interest you outside of the assigned material!

Do this:

Module	Tasks	Time to Complete	Done
Design	–		
Intro	all	2.5 hours	
UX	all	3-4 hours	
Visual Design	12, 13, 14, 17, 19	2-3 hours	
Mockups	21, 23	2 hours	
Project	We will do together	June 11	
Software Engineering	–		
Intro	all	½ hour	
Principles of Software Engineering	all	4-6 hours	
Intro to Agile Development	all	8-10 hrs	
Pseudocoding and Good Software Design	all	8-10 hrs	
Conclusion	all	1 hr	
Project	We will do together	June 11	

Projects

At the end of this section, we will meet to work on a mock client project. We will have an opportunity to put into practice our learning and see how these processes work out in the real world. This foundation knowledge will prepare us to think about how we can design software for clients.

II. Computer Science & Java

Saylor Academy CS 101

https://learn.saylor.org/course/view.php?id=6

Timeline	**–**
Time estimate to complete	**35 days @ 3-4 hours/day**
Begin	**June 15**
Project days	**June 23 & August 6**
Complete by	**August 7**

Sign-up recommended, but not required. This is a free tutorial.

We will walk through sign up and enrollment in the course during the luncheon. If you miss the luncheon, contact Amanda at amanda.walter@octanner.com if you need help registering.

Why this course?

Saylor Academy is a nonprofit academy of open learning that offers courses in numerous subjects. We will be enrolling in the CS 101: Introduction to Computer Science I. You are not required to sign up, but you will not be able to access quizzes if you are not logged in.

Do this:

Unit	Topic	Time to Complete	Done
1	Introduction	16.75 hrs	
2	Intro to OOP	10 hrs	
3	Java fundamentals	22 hrs	
4	Relational & logical operators in Java	9 hrs	
5	Control structures	20.5 hrs	
6	User defined methods	11.5 hrs	
7	Arrays	7 hrs	
8	Java I/O and exception handling	11.75 hrs	

Projects

As we learn Java, we can decide on a short project to do together- perhaps a simple game? At the end of this section we will have a project day that will present some coding-interview style problems that we will solve in small groups round-robin style!

III. JS, jQuery & Angular

Timeline	**–**
Time estimate to complete	**15 days @ 3-4 hours/day**
Begin	**August 10**
Project days	**August 27, possibly other days**
Complete by	**August 27**

We will be using Angular in the fall, so our final section will cover what you will need to be comfortable with by the time the internship starts. You will have some options on which program you learn from, choose the one that appeals to you!

If you are unfamiliar with Code School and codecademy, Code School offers short video with follow-along examples and short assessments. The videos are themed and usually interesting!

Codecademy is more reading as you do instruction, the advantage being that introductory material is written for your leisure, instead of video form.

Pluralsight offers really high quality learning content. We will make sure everyone has a 1-free-month subscription pass to use on the courses. As long as you don't activate the code before we start the 3rd module, you will not have to pay anything for Pluralsight access.

Do this:

1. **JavaScript**: *(pick one of these three)*
    1. You have experience in JS already and feel comfortable using vanilla JS in short programs.
    2. Codecademy JS track *(10 hours to complete)*

        http://www.codecademy.com/en/tracks/javascript
    3. Pluralsight "Quickstart JavaScript Vol. 1-3) *(6-8 hours to complete)*

http://www.pluralsight.com/courses/quick-start-javascript-1-1870

http://www.pluralsight.com/courses/quick-start-javascript-2-1917

http://www.pluralsight.com/courses/quick-start-javascript-3-1918

2. **jQuery**: *(pick one of these three)*
    1. codecademy jQuery track *(3 hours)*

        http://www.codecademy.com/en/tracks/jquery

    2. Code School *Try jQuery (2-3 hours)*

        https://www.codeschool.com/courses/try-jquery

    3. Pluralsight *jQuery UI (~ 2 hours)*

        http://www.pluralsight.com/courses/jqueryui

3. **JSON**:
    1. YouTube *What is JSON?* (5 minutes)

        https://www.youtube.com/watch?v=vrPBtQAxw7c

4. **Angular & REST APIs** *(do all, except where a choice is indicated)*
    1. WebConcepts video *REST API concepts and examples* (10 minutes)

        https://www.youtube.com/watch?v=7YcW25PHnAA

    2. *(pick one)*
        1. Pluralsight *Five Essential Tools for Building REST APIs* (3 hours)

            http://www.pluralsight.com/courses/five-essential-tools-building-rest-api

        2. Pluralsight *REST fundamentals* (3 hours)

            http://www.pluralsight.com/courses/rest-fundamentals

    3. *(pick one)*
        1. Codecademy Angular (5 hours) <– recommended

http://www.codecademy.com/en/learn/learn-angularjs

2. Code School Angular (5 hours) You will need to take both the free level *Shaping up with Angular.js* and *Staying Sharp with Angular.js*, for which you will need a paid membership ($29/month).

   https://www.codeschool.com/courses/shaping-up-with-angular-js
   https://www.codeschool.com/courses/staying-sharp-with-angular-js

4. Pluralsight: *TDD as a design tool* (2 hours)

   http://www.pluralsight.com/courses/tdd-as-design-tool

5. Pluralsight: *Play by Play: Learning Angular.js* (1.5 hours)

   http://www.pluralsight.com/courses/play-by-play-learning-angularjs-ken-cenerelli-john-papa

6. Pluralsight: *Angular Best Practices* (5 hours)

   http://www.pluralsight.com/courses/angular-best-practices

7. Pluralsight: *Play by Play: Angular testing* (2 hours)

   http://www.pluralsight.com/courses/play-by-play-angular-testing-papa-bell

8. Pluralsight: *Angular JS Patterns: Clean Code* (3.5 hours)

   http://www.pluralsight.com/courses/angularjs-patterns-clean-code