Urban Survival Initiative

Presents

Urban Pipeline to Software Engineering

“Becoming a Software Engineer in Six Months”

*Course Syllabus*

Table of Contents

[Introduction 4](#_Toc48821912)

[1. What is this Course? 4](#_Toc48821913)

[2. How often will we meet? 4](#_Toc48821914)

[3. Zoom Etiquette 4](#_Toc48821915)

[4. Who am I? 4](#_Toc48821916)

[5. Why is this Field in such High Demand? 5](#_Toc48821917)

[6. Who is right for this type of Career? 5](#_Toc48821918)

[7. Why Now? – Post COVID-19 5](#_Toc48821919)

[8. Getting Experience (The USI Super Store) 5](#_Toc48821920)

[9. Passing out your FREE books – Think like a Computer Scientist. 5](#_Toc48821921)

[A Class Act 6](#_Toc48821922)

[10. House Keeping 6](#_Toc48821923)

[11. The anatomy of a Class / Object 6](#_Toc48821924)

[12. More about Strings 6](#_Toc48821925)

[13. Escape Sequences 6](#_Toc48821926)

[14. Formatting Code 6](#_Toc48821927)

[15. Deeper Dive into Methods 6](#_Toc48821928)

[16. Introduction Projects 6](#_Toc48821929)

[17. Vocabulary 6](#_Toc48821930)

[18. Exercise 6](#_Toc48821931)

[Embracing the Braces 7](#_Toc48821932)

[19. House Keeping 7](#_Toc48821933)

[20. GitHub – Source Code Configuration Management 7](#_Toc48821934)

[21. Understanding Static vs. Instantiated Classes 7](#_Toc48821935)

[Variables and Operators 8](#_Toc48821936)

[22. Housekeeping – We are Recording this class 8](#_Toc48821937)

[23. Java Primitive Data Types 8](#_Toc48821938)

[24. Variables 8](#_Toc48821939)

[25. Understanding Scope 9](#_Toc48821940)

[26. Operators 9](#_Toc48821941)

[27. Google – “Java primitive data types” and study 9](#_Toc48821942)

[28. Google – “Google Java Style Guide” lean and live by it. 9](#_Toc48821943)

[29. Exercise 9](#_Toc48821944)

[30. Refresh your local repository using GitHub Desktop 10](#_Toc48821945)

[31. Commit your homework to GitHub 10](#_Toc48821946)

[32. Create a pull request with at least two reviewers. (I must be one). 10](#_Toc48821947)

[Java 1.8 Classes and Packages 11](#_Toc48821948)

[33. Housekeeping – We are Recording this class 11](#_Toc48821949)

[34. Understanding Class Method Parameters 11](#_Toc48821950)

[Java 1.8 Classes and Packages 11](#_Toc48821951)

[35. Housekeeping – We are Recording this class 11](#_Toc48821952)

[36. Recap of last training class 11](#_Toc48821953)

[37. What is the 1.8 Java Development Kit – Java SE 1.8 JDK 11](#_Toc48821954)

[38. Accessing the Classes in the Packages 12](#_Toc48821955)

[39. Access to the Code in the Book 12](#_Toc48821956)

[40. Input 12](#_Toc48821957)

[41. Homework 12](#_Toc48821958)

[Recursive Control – Loops (While, Do While, For, For Each) 13](#_Toc48821959)

[42. Housekeeping – We are Recording this class 13](#_Toc48821960)

[43. Understanding While Loops 13](#_Toc48821961)

[44. Understanding Do While Loops 13](#_Toc48821962)

[45. Understanding For Loops 13](#_Toc48821963)

[46. Understanding For Each Loops. 13](#_Toc48821964)

[47. Homework 13](#_Toc48821965)

[Accessing Data from/to Files - (Scanner | BufferedWriter) 14](#_Toc48821966)

[48. Housekeeping – We are Recording this class 14](#_Toc48821967)

[49. Writing Data to Text Files 14](#_Toc48821968)

[50. Continuing with Loops 14](#_Toc48821969)

[51. UUID/GUID – Universal Unique ID/Global Unique ID 14](#_Toc48821970)

[52. Grocery Shopping Experience – USI Super Store 14](#_Toc48821971)

[53. Homework 14](#_Toc48821972)

[Moving to Eclipse 2020 - 03 and Starting the Super Store Project 16](#_Toc48821973)

[54. Understanding Development modes 16](#_Toc48821974)

[Unified Modeling Language – UML 16](#_Toc48821975)

[55. High Level Application Design 16](#_Toc48821976)

[56. Architecture Design 16](#_Toc48821977)

[57. Process Flows 16](#_Toc48821978)

[58. Class Diagrams 16](#_Toc48821979)

[59. Data Modeling – ER diagrams 16](#_Toc48821980)

[Introduction to graphical User Interfaces - JavaFx 17](#_Toc48821981)

[60. Launch(args)/Start(…) 17](#_Toc48821982)

[61. Primary Stage 17](#_Toc48821983)

[62. Scenes 17](#_Toc48821984)

[63. Panes 17](#_Toc48821985)

[64. Labels 17](#_Toc48821986)

[65. Text Fields 17](#_Toc48821987)

[66. Buttons 17](#_Toc48821988)

[67. Event Handlers 17](#_Toc48821989)

[68. Views 17](#_Toc48821990)

[Understanding Data Storage 17](#_Toc48821991)

[69. Relational Database Management Systems – RDBMS 17](#_Toc48821992)

[70. Extensible Markup Language – XML 18](#_Toc48821993)

[71. JavaScript Object Notation – Json 18](#_Toc48821994)

[Reading and Writing Log Files – java.util.logging 18](#_Toc48821995)

[72. Purpose of logging 18](#_Toc48821996)

[73. Java 1.8 Logger Components 18](#_Toc48821997)

[74. Which events to log 18](#_Toc48821998)

[75. Which events not to log 18](#_Toc48821999)

[76. Logging best practices 18](#_Toc48822000)

[77. Event Handler 18](#_Toc48822001)

[Graduation 18](#_Toc48822002)

## Introduction

### What is this Course?

### How often will we meet?

* 1. Saturdays from 1 – 5 (2 hour Lab)
  2. Tuesday from 6 – 8
  3. Thursdays from 6 - 8
  4. When will it end?
     1. September 12th, 2020 – Graduation
     2. What happens after the course ends?
        1. Recruiters
        2. Direct Contacts with Companies
        3. Oracle Certification (Fee $240 to Oracle)
  5. What materials will I need
     1. Computer (HP/Dell – 8/16gbs RAM, 80gbs HD, Windows 10 OS)
     2. Books – Totally Free

### Zoom Etiquette ­­

* 1. Mute your sound (on entry)
  2. Video on, no spy’s (on entry)
  3. Please use your “Real Name” for our Attendance Reports, which you want on your Certificate of Completion.
  4. How and when to use Chat
  5. Raise your hand if you have questions
  6. Headphones work best
  7. Agile Teams (4 -5 Members) Breakrooms **--** Help Each Other
  8. NO Zoom Bombing!!!!!
  9. File sharing
  10. Screen Sharing and Remote Control
  11. I need a break Symbols

### Who am I?

* 1. How did I become a Software Engineer and Why?
  2. How long have I been in the industry?
  3. Where did I go to school?

### Why is this Field in such High Demand?

* 1. Salaries for Software Engineers – Glassdoor,

### Who is right for this type of Career?

### Why Now? – Post COVID-19

### Getting Experience (The USI Super Store)

### Passing out your FREE books – Think like a Computer Scientist.

#### I recommend you staying three chapters ahead of where we are in class.

#### We will be in this book until July.

#### There will be Exercises and Examines (Open Book)

## A Class Act

### House Keeping

* 1. Git Hub – Repository for Code | Books | Documents | Videos
  2. Accessing Git Hub – using your password
  3. Attendance
  4. Stay ahead by reading at least two chapters ahead of the class
  5. IDE configured? DrJava, JDoodle and Jvdroid (Google Play Store)

### The anatomy of a Class / Object

* 1. Data Members
  2. Methods
  3. Objects

### More about Strings

### Escape Sequences

### Formatting Code

### Deeper Dive into Methods

### Introduction Projects

### Vocabulary

### Exercise

## Embracing the Braces

### House Keeping

* 1. Emails – Instructors email **ibirchettsr@gmail.com**
     1. We must have your email in order to give you access to our GitHub account and Class Videos.
     2. To send out important information about the class, and changes to schedule from time to time.
  2. IDE configuration Issues?
     1. DrJava (Windows/Mac)
        1. Mac: <https://people.cs.umass.edu/~elm/Teaching/121_S12/drjava-mac.html>
        2. Windows: <http://www.drjava.org/>
     2. JDoodle -- <https://www.jdoodle.com/>
     3. Jvdroid (Google Play Store)
  3. Accessing class videos via my Google Drive.

### GitHub – Source Code Configuration Management

* 1. Creating local git directory – c:\usi-git
  2. Cloning the Software Engineering Training repository
  3. Downloading the Desktop Git app.
  4. Editing code/documents etc…
  5. Committing Changes
     1. What is the master branch?
     2. Creating your own branch. first initial, last initial, task, example: ib-helloworld.
     3. Requesting a review – Pull Request
     4. Merging the code. (I will be the gatekeeper initially)
  6. **Creating Pull Request**

### Understanding Static vs. Instantiated Classes

* 1. HelloWorld.java
  2. What is ‘new’?

## Variables and Operators

### Housekeeping – We are Recording this class

* 1. Welcome
  2. Make sure we have your email address
     1. Access to class material on GitHub
     2. Access to previous recordings of classes (so you can get caught up)
     3. Also used for class communication
     4. My email is ibirchettsr@gmailcom
  3. Extra help is available by Zoom appointment and/or after each class.
  4. Check your email to see if you receive an invitation to class videos and GitHub.
  5. This document is available in ‘Docs’ folder on GitHub in our class repositor.
  6. Turning in Homework/Exercises to the ‘homework’ folder under your own homework folder (ib-homework).
  7. Homework/Exercise should be completed before the next Saturday class.
  8. Questions/Issues?

### Java Primitive Data Types

* 1. byte
     1. [128|64|32|16|8|4|2|1] – 8 bits – On = 1, Off = 0 (2s Compliment)
  2. short
  3. Int
  4. long
  5. float
  6. double
  7. String

### Variables

* 1. Declaring Variables
  2. Initializing Variables
  3. Assignment
  4. Printing Variables

### Understanding Scope

* 1. Class Scope
  2. Method Scope
  3. Conditional Scope (if condition)
  4. Iteration Scope (For loops, while loops)

### Operators

* 1. **Arithmetic Operators**
     1. Add – ‘+’
     2. Subtract – ‘-“
     3. Multiple – ‘\*’
     4. Divide – ‘/’
     5. Remainder – ‘%’
  2. **Unary Operators**
     1. + - Positive
     2. - -- negative
     3. ++ = increment operator, increments a value by 1
     4. -- = decrement operator, decrements a value by 1
     5. != logical Complement operator, inverts the value of boolean
     6. Plus, Equals – ‘+=’
     7. Subtract Equals – ‘-=’

### Google – “Java primitive data types” and study

### Google – “Google Java Style Guide” lean and live by it.

### Exercise

* 1. Homework – Exercise
  2. Create a folder (first initial, last name) under ‘homework’ in your local c:\usi-git\homework folder in a new branch call your-initials-homework.
  3. Use the two Classes DeclaringVaiables.java and VarsExercise to complete your assignment of creating a setter and getter for each data member in DeclaringVariables.java and call them from VarsExecise.java, which is demonstrated in the existing classes.
  4. In your main class use printf to print a formatted line using the appropriate format symbol.
  5. Help: <https://alvinalexander.com/programming/printf-format-cheat-sheet/>
  6. Compile clean
  7. Do not commit any .class files.
  8. Check in .java files only.

### Refresh your local repository using GitHub Desktop

### Commit your homework to GitHub

### Create a pull request with at least two reviewers. (I must be one).

## Java 1.8 Classes and Packages

### Housekeeping – We are Recording this class

* + 1. Welcome
    2. Questions about accessing GitHub
    3. Questions about accessing Videos?
    4. Questions about Homework?
       1. How to create your homework folder.
    5. Read Chapter 3 – Input and Output this week.

### Understanding Class Method Parameters

## Java 1.8 Classes and Packages

### Housekeeping – We are Recording this class

* 1. Welcome
  2. Any Issues with accessing GitHub?
  3. Any Issues with accessing Videos?
  4. Any Issues with Homework?
     1. Any Issues with creating your homework folder.
  5. Read Chapter 3 – Input and Output this week.

### Recap of last training class

* 1. DeclaringVariables.java
     1. Methods with multiple parameters
     2. If-then-else, if-then-else-if
  2. VarExercise.java
     1. Calling methods
     2. Passing Parameters
     3. Receiving return values
     4. Printing formatted output
     5. Formatting Cheat sheet -- <https://alvinalexander.com/programming/printf-format-cheat-sheet/>

### What is the 1.8 Java Development Kit – Java SE 1.8 JDK

* + 1. What are Java Packages?
    2. Where are the Packages located?
    3. Is there documentation?
    4. Where is the documentation located?<https://docs.oracle.com/javase/8/docs/api/>

### Accessing the Classes in the Packages

* 1. Import classes in the packages

### Access to the Code in the Book

* 1. [**https://github.com/AllenDowney/ThinkJavaCode**](https://github.com/AllenDowney/ThinkJavaCode)
  2. Use it to get a greater understanding

### Input

* 1. System.in
  2. System.in.Scanner
  3. Echo.java
  4. Refactoring

### Homework

* 1. Refactor VarExercise.java to use Scanner Class to get some input values from keyboard.
  2. Call methods in DeclaringVariables.java class with inputted values.

## Recursive Control – Loops (While, Do While, For, For Each)

### Housekeeping – We are Recording this class

* + 1. Welcome
    2. Any Issues with accessing GitHub?
    3. Any Issues with accessing Videos?
    4. Any Issues with Homework?
       1. Any Issues with creating your homework folder.
    5. Read Chapter 3 – Input and Output this week.

### Understanding While Loops

* + 1. Break command
    2. Continue command
    3. Reading from a file with a while loop

### Understanding Do While Loops

### Understanding For Loops

### Understanding For Each Loops.

### Homework

* + 1. Please continue to implement WhileTest.java and WhileExercise.java in my homework folder: ib-homework

## Accessing Data from/to Files - (Scanner | BufferedWriter)

### Housekeeping – We are Recording this class

* + 1. Welcome
    2. Any Issues with accessing GitHub?
       1. **Desktop GitHub - Refresh Daily**
    3. Any Issues with accessing Videos?
    4. Any Issues with Homework?
       1. Homework due Saturday.
       2. No pull request found for homework.
    5. Still in Chapter 7 While Loops.

### Writing Data to Text Files

* + 1. Creating Configuration files.
    2. Reading Configuration files.

### Continuing with Loops

* + 1. Do {} While () Loops
    2. For Loops
    3. For Each Loops

### UUID/GUID – Universal Unique ID/Global Unique ID

### Grocery Shopping Experience – USI Super Store

* 1. Welcome
     1. Greeting Class (includes main method)
     2. Membership Class
  2. Shopping Cart Class
     1. Shopping List Class
     2. Add To Cart Class
     3. Check Out Class
  3. Product List Class
     1. List by Department Class
     2. List by Product Class
  4. Create supporting text files

### Homework

* 1. Create an application that will do the following
     1. Create a main class
     2. Create a class that uses a while loop to collect data until the word “end” is entered.
        1. Write that data into a text file, one line at a time.
        2. When “end” is entered do not write it to file, but break from the while and close the file.
  2. Verify your content by using sublime or Notepad++ text editor to review.
  3. Create a pull request for the data.

## Moving to Eclipse 2020 - 03 and Starting the Super Store Project

### Understanding Development modes

#### Debug Mode - Debugging

#### Development Mode

#### Running Applications

#### Understanding the Console output

#### Using the Project Explorer pane

#### Using the Outline pang

#### Using the Problems Pane

#### Using Search Features

#### What are Breakpoints

## Unified Modeling Language – UML

### High Level Application Design

### Architecture Design

### Process Flows

### Class Diagrams

### Data Modeling – ER diagrams

## Introduction to graphical User Interfaces - JavaFx

### Launch(args)/Start(…)

### Primary Stage

### Scenes

### Panes

#### HBox

#### HBox

#### Grid Pane

### Labels

### Text Fields

### Buttons

#### Combo Boxes

### Event Handlers

#### Actions

#### Window

### Views

#### Image Views

#### Media Views (Videos)

## Understanding Data Storage

### Relational Database Management Systems – RDBMS

#### Tables

#### Columns

#### DB Types

#### Data Parent Child Relationships -- Table Joins

#### Indexes

#### SQL Syntax

#### Queries

#### Schemas

#### Stored Procedures

### Extensible Markup Language – XML

### JavaScript Object Notation – Json

## Reading and Writing Log Files – java.util.logging

### Purpose of logging

### Java 1.8 Logger Components

### Which events to log

#### SEVERE (highest level)

#### WARNING

#### INFO

#### CONFIG

#### FINE

#### FINER

#### FINEST (lowest level)

### Which events not to log

### Logging best practices

### Event Handler

## Graduation