*Course Syllabus*

Day #1

Introduction

1. **Introduction**
2. **Welcome**
   1. What is this Course?
   2. How often will we meet?
      1. Saturdays from 1 – 5 (2 hour Lab)
      2. Tuesday from 6 – 8
      3. Thursdays from 6 - 8
   3. 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)
   4. What materials will I need
      1. Computer (HP/Dell – 8/16gbs RAM, 80gbs HD, Windows 10 OS)
      2. Books – Totally Free
3. **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
4. **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?
   4. Who is on the USI team?
      1. **Roxanne Earnest** – IT Administration and Support
      2. **Dr. Colleen Birchett** – The Dean of Student Development and Communications
5. **Why is this Field in such High Demand?**
   1. Salaries for Software Engineers – Glassdoor,
6. **Who is right for this type of Career?**
7. **Why Now – Post COVID-19**
8. **Getting Experience (The Contact Tracing Software Development Project)**
9. **Passing out your books – Think like a Computer Scientist.**
10. **I HIGHLY recommend staying three chapters ahead of where we are in class.**
11. **We will be in this book until July.**
12. **There will be Exercises and Examines (Open Book)**
13. **TIME CHECK – Break**

Day #2

A Class Act

1. **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 one chapter ahead of the class
   5. IDE configured? DrJava, JDoodle and Jvdroid (Google Play Store)
2. **The anatomy of a Class / Object**
   1. Data Members
   2. Methods
   3. Objects
3. **More about Strings**
4. **Escape Sequences**
5. **Formatting Code**
6. **Deeper Dive into Methods**
7. **Introduction Projects**
8. **Vocabulary**
9. **Exercise**

Day #3

Embracing the Braces

1. **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 configured 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.
2. **GitHub**
   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)
3. **Understanding Static vs. Instantiated Classes** 
   1. HelloWorld.java
   2. What is ‘new’?

Day #4

Variables and Operators

(Chapter 2 in Class Book)

1. **Housekeeping – We are Recording this class**
   1. **Welcome – Happy Mother’s Day**
   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?**
2. **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**
3. **Variables**
   1. **Declaring Variables**
   2. **Initializing Variables**
   3. **Assignment**
   4. **Printing Variables**
4. **Understanding Scope**
   1. **Class Scope**
   2. **Method Scope**
   3. **Conditional Scope (if condition)**
   4. **Iteration Scope (For loops, while loops)**
5. **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 – ‘-=’
6. **Google – “Java primitive data types” and study**
7. **Google – “Google Java Style Guide” lean and live by it.**
8. **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.
9. **Refresh your local repository using GitHub Desktop**
10. **Commit your homework to GitHub**
11. **Create a pull request with at least two reviewers. (I must be one).**

Day 5/6

Java 1.8 Classes and Packages

1. **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.
2. **Understanding Class Method Parameters**

Day 7/8

Java 1.8 Classes and Packages

(Input and Output, Chapter 3)

1. **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.
2. **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/>
3. **What is the 1.8 Java Development Kit – Java SE 1.8 JDK**
   1. What are Java Packages?
   2. Where are they Packages located?
   3. Is there documentation?
   4. Where is the documentation located?<https://docs.oracle.com/javase/8/docs/api/>
4. **Accessing the Classes in the Packages**
   1. Import classes in the packages
5. **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
6. **Input**
   1. System.in
   2. System.in.Scanner
   3. Echo.java
   4. Refactoring
7. **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.

Day 9 /10/11

Recursive Control -- Loops

(While, Do While, For, For Each)

1. **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.
   6. NO CLASS THIS WEEKEND
2. Understanding While Loops
   1. Break command
   2. Continue command
   3. Reading from a file with a while loop
3. Understanding Do While Loops
4. Understanding For Loops
5. Understanding For Each Loops.
6. **Homework**
   1. Please continue to implement WhileTest.java and WhileExercise.java in my homework folder: ib-homework

Day 12/13

Accessing Data from Files

(Scanner | BufferedWriter)

1. **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 today.
      2. No pull request found for homework.
   5. Still in Chapter 7 While Loops.
2. **Writing Data to Text Files** 
   1. Creating Configuration files.
   2. Reading Configuration files.
3. **Continuing with Loops**
   1. Do {} While () Loops
   2. For Loops
   3. For Each Loops
4. **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 work “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 you content by using sublime or NotePad++ text editor to review.
   3. Create a pull request for the data.