Day #1

Course Syllabus

Introduction

* Welcome
  + What is this Course?
  + How often will we meet?
    - Saturdays from 1 – 5 (2 hour Lab)
    - Tuesday from 6 – 8
    - Thursdays from 6 - 8
  + When will it end?
    - September 12th, 2020 – Graduation
    - What happens after the course ends?
      * Recruiters
      * Direct Contacts with Companies
      * Oracle Certification (Fee $240 to Oracle)
  + What materials will I need
    - Computer (HP/Dell – 8/16gbs RAM, 80gbs HD, Windows 10 OS)
    - Books – Totally Free
* Zoom Etiquette ­­
  + Mute your sound (on entry)
  + Video on, no spy’s (on entry)
  + Please use your “Real Name” for our Attendance Reports, which you want on your Certificate of Completion.
  + How and when to use Chat
  + Raise your hand if you have questions
  + Headphones work best
  + **Agile Teams (4 -5 Members) Breakrooms --** Help Each Other
  + NO Zoom Bombing!!!!!
  + File sharing
  + Screen Sharing and Remote Control
  + I need a break Symbols
* Who am I?
  + How did I become a Software Engineer and Why?
  + How long have I been in the industry?
  + Where did I go to school?
  + Who is on the USI team?
    - **Roxanne Earnest** – IT Administration and Support
    - **Dr. Colleen Birchett** – The Dean of Student Development and Communications
* Why is this Field in such High Demand?
  + Salaries for Software Engineers – Glassdoor,
* Who is right for this type of Career?
* Why Now – Post COVID-19
* Getting Experience (The Contact Tracing Software Development Project)

# Think like a Computer Scientist the book.

* Passing out your books.
* I HIGHLY recommend 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)

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).**