**HW 1 – CS 4321, Fall 2015**

|  |  |
| --- | --- |
| Name: | Tyler Cabutto |

**Overview**

<<<Remove this section before submitting>>>

You will watch the videos for the first lesson from the Udacity course, Software Development Process and answer the questions below. **Type your answers, print them, and bring them to class on Wednesday, August 19.**

Directions:

1. Go to <https://www.udacity.com/>
2. Sign Up
3. Choose: Catalog, Georgia Tech Masters in CS, Software Development Process
4. Choose: Access Course Materials.
5. It will probably be showing you the *Dashboard*. Choose the first link: “Introduction and Overview”



1. Watch all the videos in the, “Introduction and Overview”. It is a total of 22 minutes.
2. **Type your answers to the questions below, print them, and bring them to class on Wednesday, August 19.**

**Questions – Lesson 1: Introduction and Overview**

1. What is software engineering? List as many facets as possible.

* The discipline that investigates programming development, and what would be the best way to code a programming.
* Systematic application of methods to build software in a rigorous way. Involves building the system, meeting requirements, and meet the stakeholder needs.
* Process of creating software of engineering principals.
* Entire life cycle, and also about the quality about the software. This way you will think about long after it is shipped.
* Improves better software and quality and productivity.
* Set of activities that one engages in when building software systems or products. It is a venue creating activity, and social processes.
* People working together to make a large complex system.
* Envisioning and realizing valuable new functions, and having confidence that it has all attributes needed to be successful, for the software components of engineering systems, and for the system overall, given that it is reliant on the software.
* Art and practice of building software system.
* A set of methods and principles and techniques that we have developed to enable us to engineer or build large software system that is more complex for an engineer, or small team of engineers to understand and construct and maintain.
* Building and constructing at a large scale, high quality systems.
* The engineering discipline of developing software based systems, usually embedded into larger systems composed of hardware and even humans, also business processes, and processes in general.
* Mostly being able to program, and being able to put together big systems together to actually work.

1. Why is it important?

* Because software is important to our everyday lives now and we need engineers to ensure that need is met.
* Because software is pervasive in all industry sectors and therefore systems must be reliable, safe, and secure.
* Because if you don’t think about how you’re building this system and how you’re trading off different aspects such as performance, scalability, and reliability, then it’s going to end up braking or not lasting long, or not doing what you want, or costly.
* If it is not done with principles, everything will fail, including the user.
* Because you are aiming for two goals: productivity, and performance (higher quality).
* We use software in everyday life. It’s everywhere now.

1. What are the three causes of the software crisis?
2. Software had a higher demand than hardware.
3. Increasing product complexity.
4. Developer’s productivity growth.
5. What event (and when) is considered the birth of software engineering?

* NATO software engineering conference in January, 1969.

1. Define software development and explain how that leads to a software process?

* Software development is going from an abstract idea in someone’s head such as the customer’s head, to a concrete system that actually implements that idea and hopefully works. It leads to it by breaking it down to small systematic, or formal steps to reach the overall goal.

1. How many lines of code do professional developers average per day? Why is the number not larger?

* 50-100 lines of code a day. Because coding is not everything while developing a system.

1. What are the five phases that characterize all software processes?
2. Requirements Engineering.
3. Design.
4. Implementation.
5. Verification and Validation.
6. Maintenance.
7. What are three areas where tools of the trade have helped close the gap between software size and complexity and developers’ productivity?

* Development 🡪 Punch cards 🡪 IDE’s
* Language 🡪 Machine Code 🡪 High-level languages.
* Debugging 🡪 Print lines 🡪 Symbolic debuggers.