## Introduction

* Define software engineering.
* What is the difference between generic and custom software?
* Differentiate between depth of complexity and breadth of complexity
* Be able to calculate the number of communication paths given the number of people
* List some reasons for software project success
* What is a successful software project?

## Software Process Models

* What are the typical software lifecycle activities?
* What is a software process model?
* Describe the following software process models
  + The waterfall model
  + Incremental model (advantages)
  + Spiral model (know the four sectors in Boehm's spiral model)
* Agile methods: what are the key features, why was it developed and what is its purpose?
* Pair programming: what is it and why use it?

## Requirements

* Know why requirements engineering is important.
* Differentiate between functional and non-functional requirements. Given a scenario identify several of each.
* Given a scenario, be able to draw a use case diagram.

## Project Management

* Estimating the size of a project
  + LOC vs. function points
  + Know what they are and the benefits and drawbacks of each
* Be able to draw an activity diagram, find the critical path and compute slack times

## Version Control

* Why does teamwork increase the complexity of version control?
* What is the major difference between distributed version control systems different and centralized version control systems?