**1. 20 points) List the three types of development testing that were mentioned in lecture and**

**provide a brief (1-2 sentences) description of each.**

Development – This type of testing is going to be done by the development team, the ones who are writing the code/building the system. They are testing that the code functions and do what it is designed for.

Release – Release testing is done by hired people, who are given a version of the system and tasked with testing certain preset requirements that the system should be able to do. They go through the system and make sure everything works, much like how this semester, the grader went through and graded our project based on requirements.

User – Testing is made available to public users (not hired to test specific parts of the system). The user is available to go through all parts of the system and just mess around with it, see if they can find any bugs, report them (if they want, not required), or just simply if they like the product, and maybe give feedback as to why or why not and help the development team work to fix the cons and keep the pros.

**2. 20 points) Considering our lectures on clean code, identify what is wrong with the following**

**segment of code. To identify please indicate (a) what part of the code you are discussing, (b) the**

**problem that exists, and (c) how you would fix it.**

**◦ public List<int[]> getThem() {**

**List<int[]> list1 = new ArrayList<int[]>();**

**for (int[] x : theList)**

**if (x[0] == 4)**

**list1.add(x);**

**return list1;**

**}**

Problem 1: a) The naming of the function, the array list, and the variable x.

b) The name of the function should help give context to what the function is doing and would help us understand the naming of the array list and variables inside the function. The array list name should not include the literal value “1” in its name, and it could be telling us what kind of list. The name of the variable x defined in the for each loop once again does not help give context to what this function is doing to outside users other than the author.

c) What is “Them” in the function name getThem, instead write out what we are “getting” so if gives context to rest of the naming within the function. The array name list1 is not good because it should not contain the actual value ‘1’. Instead call it listOne. For the last complaint, the name “x” should be changed to something more practical so the next user will understand what the function is doing. Fixing the function name would help this problem out the most, giving context to rest of everything within it.

Problem 2: a) Formatting within the function

b) The if statement and the for each loop are not using curly braces at all to define their range.

c) Put beginning and end braces for both the if statement and the for each loop.

**3. 20 points) Also referencing our lectures on clean code, describe class cohesion making sure to**

**provide the characteristics of a cohesive class.**

A good class in software engineering should be cohesive. This means that it should be small and concise. Small in the sense not of amount of code of functions, but small in the sense of number of responsibilities a class is taking up. Each class should try to follow the Single Responsibility Principle (SRP). The responsibility in this term basically means purpose. Clean code principles state that many different smaller, SRP classes are better than fewer but bigger and unorganized classes. This all ties into class cohesion. The more a class follows the SRP rule, the smaller number of instance variables a class should have. Class cohesion can be measured by the percentage of methods in a class that are manipulating the instance variables of it class. For better cohesion, less is more.

**4. 20 points) In the ACM webinar "Stranger than Fiction Case Studies in Software Engineering**

**Judgment" - [https://www.youtube.com/embed/PFcHX0Menno], Steve McConnell discussed**

**Bloom's Taxonomy with respect to Software Engineering. Specifically he considered Analysis**

**to be an "over-developed muscle" of a typical developer. What level of Bloom's Taxonomy does**

**Mr. McConnell recommend software engineers develop and, very simply, why?**

Mr. McConnell believes and recommends that software engineers should try to develop judgement(evaluation). He believes and provides multiple case studies using the four-factor model that give evidence that even highly intelligent software engineers have a deficiency in judgement. Mr. McConnell states that this deficiency in judgement often in software leads to **gross** lapses in judgement that lead to major unsuccessful projects. He states that judgement is the highest order thinking skill in Bloom’s Taxonomy pyramid and therefore relies and depends on the skills below it. He states in software, there is multiple parts in project dynamics where you use judgement/evaluation skills. He claims (and shows with the case studies) that software engineers often get analysis paralysis and even in hindsight, never realize they actually had bad judgement and made the wrong decision early on that had been the start of the downfall of the project.

**5. 20 points) Ethics short answer. Please reference [https://ethics.acm.org/code-of-ethics/softwareengineering-code/] as you answer the following questions, providing brief responses (1-2**

**sentences) for each.**

**◦ 5 points) Principle 2 item 2.01 and Principle 3 item 3.04 deal with qualification and**

**competency. Why do you think it is important to accurately represent yourself as a**

**developer?**

I believe it is important because if you lie or are silent about what you can or cannot do, you may prevent the project from being a success. This could end up wasting a lot of time and money and your employer may fire you and this will not look good to other employers. Also, depending on the project or product you are working with, you are hindering the project and could hurt people who are relying on this product or design for health or day to day needs.

**◦ 5 points) Principle 4, item 4.02 provides criteria to be used for deciding when to endorse**

**documents. Provide an example (may be hypothetical) of when you should not endorse a**

**document.**

Maybe you are asked by a peer of yours who has created a recommendation report that says his design or model could help nurses or doctors more efficiently. If you don’t actually read the document, or don’t look at it with your best unbiased judgement, and just endorse it because it is your friend or peer who wants you to endorse it, this design or model could actually be very bad and then you could be hurting healthcare workers and then you look bad for endorsing a bad practice that should never got off the ground.

**◦ 5 points) If you built an app that accessed a user's contacts and made use of them in some**

**way without disclosing this to the user which item(s) would this violate.**

1.02. Moderate the interests of the software engineer, the employer, the client and the users with the public good.

1.03. Approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy or harm the environment. The ultimate effect of the work should be to the public good.

1.04. Disclose to appropriate persons or authorities any actual or potential danger to the user, the public, or the environment, that they reasonably believe to be associated with software or related documents.

1.05. Cooperate in efforts to address matters of grave public concern caused by software, its installation, maintenance, support or documentation.

3.01. Strive for high quality, acceptable cost and a reasonable schedule, ensuring significant tradeoffs are clear to and accepted by the employer and the client, and are available for consideration by the user and the public.

6.06. Obey all laws governing their work, unless, in exceptional circumstances, such compliance is inconsistent with the public interest.

**◦ 5 points) Pick another item, indicating which you have chosen, and discuss it briefly. (This**

**could be further elaboration, why it is important, example scenarios where it might be**

**applied, obvious violations, etc...)**

2.08. Accept no outside work detrimental to the work they perform for their primary employer.

An example of this could be that you are working for a software company and the work is getting kind of slow. You decide to look for other projects online that you could apply to or start to work on to earn more money and be more active with your time. If you do not disclose this to your employer or you do and they tell you no, maybe the work at your primary employer one day gets a contract from the government to work on this other project. Now you have two different projects/employers you are working for and now not enough time to put in good quality work for both. In this case, you are being detrimental to your work now for both employers. This is significant because you are now making your teammates look bad, you look bad, and your employers look bad and work is not getting done like it should be. This is different from having a hobby or working on your own small project that does not have some sort of a deadline and would not end up being detrimental to your work for your primary employer.