1. Question one

There are ways in which the typical approaches to problem solving in computer science are limiting in regards to “positive professional judgment”. Positive professional judgment includes but is not limited to consideration of applications being developed and their impact on the world. Such judgment is also extended to the way in which software is developed, if software is developed sloppily, it could leave room for outsiders to make use of said software in a malicious manner. Software should be developed in a robust fashion, lest a user accidentally find an issue, leading to further problems in a program or on a given system.

A typical approach involves using development cycles, such as scrum or the unified process to develop robust and secure software. However, these cycles are limited due to their format and constraints, not all development cycles ensure that software is properly tested and secure. Such development cycles should be extended to incorporate thorough testing, with checks and balances for programmers, ensuring that no malicious software is accidently released. There are several limitations to the typical software development approach; for instance, being bound to a specific approach can make fixing certain issues problematic. An instance being the spiral model that would require the issue be dealt with at a later date. Ideally, there would be a group of people assigned to verify that a given software solution is designed to limit possible misuse of said software. Professional writing involves doing everything one can do in order to produce software that is ethically sound, such as keeping up with the latest coding practices and languages.

In the lecture there are four main groups of programmers, one is malicious programmers; the second is programmers that do all that they can to ensure safe software; the third is professionals that do the least possible work, leaving room for exploits to exist (noted in the lecture as “competent slum lords”); the final group of programmers are simply incompetent programmers, ones who can unwittingly make malicious software. Ideally, every programmer should be in the second group, doing all that they can to create safe and useful software. These individuals are the true professionals, ones who care about the computing world and who said world effects. In order to prevent becoming an unethical programmer, there are several key steps that must be taken. One noted step is “self-image” the thought that you as a programmer are making something to better society and not to exploit it. Another step is to be “conceptually bounded”, having a mind set that causes the programmer to be responsibly bound to make secure software. Finally, ethical binding must be fulfilled, which binds a programmer to not make malicious or sloppily designed software.

1. Question two

Software professional licensing is the process by which a company would license a software professional who is certified to be ethically sound. Such an idea makes sense as not all programmers can be trusted or have the requisite skill to work on a given project. As previously stated, a certified programmer will be licensed saying that they are competent and trustworthy to work on a project.   
Such a certification could potentially prevent loss of information due to outside penetration, as well as halt lawsuits, which would be damning to a company’s image. Furthermore, the workplace would be filled with trusted and talented employees allowing for the development of stronger software. The idea of professional licensing could impact my, as well as any student’s career, as, if such a process becomes standard practice, I will need to be certified in order to receive a well paid job. Such measures are necessary, due to the fact that, for the most part, a company will hire a certified programmer over an uncertified one a majority of the time. There are many positive outcomes of hiring certified professionals, including but not limited to faster development cycles due to knowledgeable employees; lower cost due to efficiency; piece of mind, knowing that employees will leave a few gaps in a system as possible as well as dedication, which any licensed employee should have. There are few concerns with certified employees, which is the rational behind hiring them. If a company is given the choice to hire an uncertified employee and a certified one, the certified employee will be picked more often than not. Such an outcome is due to a social effect where those with certification are better than those without. This effect can be seen in the esteem of students coming from a college with a strong reputation often being picked over those with lesser degrees. A professional writing style is one in which all pieces of software are written and tested in such a way that they both do their job and are secure from being tampered with, leaving holes in a system. Furthermore, professional writing is never intentionally malicious, meaning that a programmer will not write something, which could cause harm to others. That being said, there are many examples of unprofessional code writing, most notably, from the NSA, who set up the PRISM system to spy on citizens, which, could quite easily be seen as malicious software put to use on a wide scale.

There are concerns I have about licensing developers namely that it could narrow the market for the computer industry. Not everyone can necessarily afford to get a license. Such a fact could ruin the development community, as we know it, first time or self taught programmers would have a difficult time finding a job in a major company despite their skill. Such a thought would be terrifying, it could ruin the face of the development community as it is currently known as well as possibly put a number of programmers out of work. What I am trying to say is that licensing could be used as a means of discrimination for some programmers, something that is not acceptable.