Spring 2021

Name ______________________________

CS456 - Exam 1, Part I - DUE Monday 2/22 by 11:59pm
(48 Points)

Directions

- You may spend as much time prior to writing your responses as you wish *organizing* your thoughts. This includes making notes, sketching outlines, etc. but should not involve discussing the questions with anyone other than the instructor or writing any portions of your final answers. Answers will be graded based upon the recognition of the validity of differing viewpoints (and the consequences of each position) and a *thoroughly justified* discussion of your opinions on the issue **utilizing the resources from class. Be sure to cite any class resources you use in your discussion** via [#] notation and a works cited page. You *should not* use any other resources except, if desired, for strictly factual statistics.
- You are allowed **1 hour** to write your final answers. You may divide up the alloted time into parts, but you must complete any questions you begin within a given session. You may then take an additional 15 min. to *proofread* your responses and correct any spelling, grammar, etc.
- Answers for each question should be **no more than one page**. Each answer should **begin on a new page** and use 10-point font with double spacing **- a 10% penalty will be deducted for improper formatting**. Include a separate works cited page at the end numbering the references used in your answer. **Upload** a single .pdf (named *username*-CS456-Exam1a.pdf) with your submission answers to Canvas by 11:59pm Monday, February 22.

1.(24 points) **Professional Software Development**

Your CS education at YCP is aimed at giving you the tools necessary to be a professional in the computing industry who employs good software development practices, including agile development methodologies with sufficient testing procedures, in an attempt to mitigate introducing bugs and deliver quality software. However even with your education, as practicing software engineers you will have to accept the fact that all but the most trivial code you write will contain bugs - particularly when it is just one piece of a larger project. Additionally, often management is hesitant to provide the necessary time and resources to thoroughly test software in order to meet release deadlines and customer price points. As software continues to become more complex and increasingly incorporated into critical systems, particularly in multitasking and distributed environments, pinpointing which component may have caused a failure can be nearly impossible. Yet, failure of these systems can be a minor inconvienence at best to significant financial loss and even personal harm at worst when they are integral to products and services we use (and even rely on) on a daily basis.

Given this ever growing dependency on and impact failures can have, do you think professional organizations, such as the ACM and IEEE, or the government have a responsibility to regulate the software industry? Would there be advantages or disadvantages to instituting professional standards, including possible licensing, contingent on following a Code of Ethics for the computing industry, and if so, by whom? Could such standards be useful in limiting liability and/or improve the societal perception of software developers as more than just hackers or IT? What role do you feel these organizations should play in protecting the interests of both consumers and corporations through possible legislation to hold software companies accountable for their products, similar to warranties in other industries? Does the degree of accountablility depend on the type of software, and what type of compensation, if any, should there be for software failures? Is the issue more substantial than just more exhaustive testing when it comes to *system* level failures? Do we have any responsibility if we build software to meet a specification that may contain inherent risks, or is it our job to also incorporate safety mechanisms to minimize damages?

2.(24 points) **Intellectual Property**

With the easy redistribution of copyrighted material on the Internet, it is difficult for content creators to monitor all the different websites for copyright violations. A big push to address this issue, spearheaded by the EU through the DSM Directive, is aimed at trying to protect copyrighted material by making sites that *distribute* content *proactively* responsible for assuring that there is not illegal copyright violations (Article 13) posted on their sites. Furthermore, there is a growing push to reduce the ”safe harbor” protections that content hosting sites have used to avoid liability so long as they respond promptly to notifications of copyright violations. While the DSM Directive may not be perfect in its current form, especially with respect to fair use, discuss whether or not its intent is justified and what modifications to its implementation you would suggest to fairly protect the value and compensation content creators deserve for their work.

In regards to software, one movement in the software community promotes the idea that all software should be open source, i.e. “freely” available. One common license is the MIT version:

https://mit-license.org

Discuss the ramifications to software creation, use, and distribution if all software was governed by the MIT license. Present the advantages and disadvantages if such a policy would be legislated as the intellectual property statute for the entire software industry from the standpoint of economic feasibility, societal benefit, and technological innovation. *Should* software developers and/or companies have a right to current intellectual property protections, e.g. copyrights and patents, or should an alternative IP category be created for software? What protections would this new IP category grant/deny to the developer, who would grant the protections, and who would enforce them? Should different aspects of software, e.g. the conceptual framework (design and/or API interface), source code (implementation), object code (executables), graphical interface layout (visuals), etc. be protected by different categories or even at all? How could this new category(s) circumvent the complicated cross licensing problems that plague the current patent system or does creating a fair and equitable patent commons address any of these downsides?