**Software Entropy**

1. Which IDE did you use for the last Python course? Will you use it again? Why or why not?
   * During the last course and over the past 8 years, I’ve chosen to use Visual Studio Code for Development purposes because of the lightweight features and scalability of the software. Visual Studio Code has excelled in being easy to use and flexible with different projects and application code.
2. What resource did/will you use the most for programming issues? Why this one?
   * During the last course I used the resource knowledge base within Bellevue, YouTube, stack overflow, reddit, and google to research general information on topics. Out of this bundle, I used information on YouTube the most because of the visualize representation and walk-through techniques. Plus, Bellevue has referenced material on YouTube for these reasons.
3. Using the ***Pragmatic Programmer*** reading assignment, select one (1) topic and complete the following:
   * Why did you select this topic?
     1. I select software entropy due to its surprising effect of a developers success overtime.
   * Summarize the main points (in your own words) of that topic in three or four sentences.
     1. Software Entropy – I chose software entropy for my topic because I’ve experienced how the gradual effects of unmaintained code can cause security issues to evolve within your application. Developing a fully functional application can be challenging and rewarding for a developer. However, the real challenge begins after development, as software entropy increases over time. Maintaining the application becomes an ongoing task that may go unnoticed without years of experience.
   * Find at least one additional resource (video, book, article, website, etc.) that supports your summary. Write an additional 1-2 sentences. Include a link to that resource.
     1. <https://www.youtube.com/watch?v=U3sWzKnMIbk>
     2. I found this video about How to Reverse Software Entropy by Saleem Siddiqui, interesting. He talked about the entropy of an isolated software system must always increase until it reaches equilibrium at maximum entropy. This is specifically relatable to how a software will naturally degrade without maintenance. A way to measure entropy is the number of files changed with each commit in the source control system. Another way to detect entropy is the ratio of unexpected problems to the scope of work in an iteration.
        + Large number of problems = High degree of disorder in a software system