Assignment: Week 2 Articulate: *Version Control*

Course: Programming with Classes (CSE 210; Section A5)

Professor: Eric Carter

Submission Due Date: Saturday, Apr 29

Submitted By: Andrew Palmer

Student ID: 886695731

**What is version control and why is it important?**

Version Control is a system of ordering saved program files such that those files may be accessed through novel ways, and that that system inherently benefits the programmers using said files by a number of different means. These benefits include an easy way for making backup copies of files; the ability to automatically keep a history of changes made; the ability to view these changes; the ability to collaborate with team members; and the ability to do all this within an environment of experimentation. To elaborate further on these concepts, let’s walk-through an example of how version control could be used, particularly highlighting the elements of collaboration and experimentation.

For our example, suppose a work team was tasked with designing a program that needed to render a new product design for the company they work for. Suppose the product was somewhat complicated in nature, and therefore the program itself was too, thus necessitating a group effort.

This type of situation shines a light on how version control systems help. If, for instance, that same task was put on the shoulders of a single individual, a number of problems might occur. These include – but are not limited to - the possibility of not completing the work on time, and the increased likelihood of errors, given that there’s less oversight throughout the development process.

To highlight the value of version control further, let’s continue with the example. Suppose one of the team members had the assignment to work on one element of the needed program. Assume their role was to ‘create functions that allowed a user to input different product preferences, and then to translate those preferences into internal company codes that other team members would use as parameters in functions they are to develop’.

Based on this added information, we can see that indeed there very likely will need to be a high degree of collaboration in this work. But let’s also assume that there’s a deadline that is arriving imminently, and each member of the team wants to get to work soon. So, rather than each person waiting in a virtual line, the programmer we’ve described above does a simple version of their program and uploads it to a Git repository using the command *git add –all*. (Git is one option for version control; it uses a *Distributed Version Control System* (DVCS) architecture).

To sum this up, the advantage of the described programmer quickly writing some initial code and then saving the file in this means is that the other team members can then use said code as the inputs for their required functions. Furthermore, as this initial code is improved upon, it will automatically become part of the ongoing file that effectively everyone is *virtually* working on together.

Yet that’s not all. Suppose there was a problem in the file, either as it was developed or afterward. Because it was made and housed in a version control environment, its different iterations were all saved, and team members can all look back through it to see where the work went awry.

Further to this final point, multiple individuals could easily be assigned the same task within such a project. If one person attempts to write the code one way, it may work, but it may not be as helpful as another person’s work. Thus, in this arena of collaboration and experimentation all benefit, and the best solution wins out.