**CS 623 Summer I**

**Exercise on GitHub and Git**

**Part 3: Q&A**

1. What is GitHub?

GitHub is a website that allows people all over the world to push their code to a remote server, pull other projects or changes to codebases they’re collaborating on using Git. GitHub makes it easy to work on a project together by acting as the central repository.

1. When was it created?

GitHub was created/founded in February 2008 and launched in April 2008. The first line of code for GitHub was written in October 2007.

(Source: <https://github.com/about/milestones>)

1. Why?

GitHub was built to free developers “from the main line of development” and allow for experimentation. GitHub is powered by Git. Previous to Git most people used Subversion, where any changes committed are pushed to the main branch.

(Source: <https://www.infoworld.com/article/2615989/github-ceo--we-re-helping-software-eat-the-world.html>)

1. By who?

GitHub was created by Chris Wanstrath, PJ Hyett, Tom Preston-Werner, and Scott Chacon.

(Source: <https://github.com/about/facts>)

1. What similar platforms exist?

At the time of writing, similar platforms include Bitbucket and GitLab. Bitbucket supports Mercurial and Git for version control. GitLab supports Git and has built-in support for Continuous Integration/Continuous Delivery, which GitHub doesn’t.

1. Why would you use such a platform?

In general, you would use these platforms if you wanted to remotely store code for a project you’ve worked on, and/or to collaborate on projects with others (open source or closed) and maintain version control for tracking changes, rolling back, etc.

**Part 4: Git Tutorial Notes (Things Learned)**

Git-it is a tutorial with a lot of great information on Git and GitHub. Before opening up the tutorial, the process felt a little overwhelming because I wasn’t sure what I was getting into but once I launched the app, everything went smoothly.

When adding your GitHub username to your Git configuration, the username is case sensitive. Didn’t know this!

git push <REMOTENAME> --delete <BRANCHNAME> will come in handy! I often forget to delete remote branches when I merge through terminal and usually end up doing this through GitHub instead.