In the first module, you learned about how different version control systems and effective software development workflows enable modern software developers to collaborate across the world without messing up each other's code. You gained knowledge about the history of version control and know how version control or subversion is used to bring order to the chaos of massive software projects, they have the potential for mistakes and bugs. Next, you learned more about the various systems, tools and methodologies that are leveraged by software developers to collaborate successfully as part of a global team. You've explored how to resolve conflicts in Git, and that version control plays a crucial part in the development of software. You then moved down to investigate the difference between staging and production, and that a staging environment should mimic your production environment. In Module 2, you learned about how to use the command line to execute commands in Linux. You were introduced to what the command line is and learn to use commands that traverse, create, rename, and delete files on your hard drive. Then you learned how easy it is to use piping and redirection to create powerful workflows that will automate your work, saving you time and effort. Finally, you explored the command line further, discovering standard input, output streams, flags that can be used to change the behavior of a command and grab. In Module 3, you developed a strong conceptual understanding of the Git technology and how it is used in software development projects to manage team files. First, you learned how to install Git on various operating systems, and then how to connect to Git Hub via HTTPS and SSH before creating a GitHub account. Next, you gained a practical understanding of how Git works, including creating and cloning a repository, add, commit, push and pull. You also explored how to use a repository in some concepts associated with workflows such as branches, blame, and forking. Finally, the ungraded lab is an opportunity to complete a practical version control exercise by forking a repository, creating a branch and committing a change. It also includes staging your changes and opening a pull request with a source repo.