# Overview of the Assignment:

Please analyze the described GIT process and respond to the following six questions.

*Be sure to submit your completed assignment via Blackboard using the Assignment tool.*

|  |
| --- |
| Consider the sequence of ten steps, while collaborating with others to develop software, |
| 1. create your own branch |
| 1. execute commits - make changes on your branch |
| 1. execute unit test on your branch |
| 1. conduct peer review of changes on your branch |
| 1. make additional changes on your branch |
| 1. perform rebase from master into your branch |
| 1. resolve any conflicts that might occur from rebase |
| 1. attempt performing an auto-merge from branch into master |
| 1. if auto-merge does not go thru, perform manual merge |
| 1. resolve any conflicts that might occur from merge |
| **Write few lines** responding to each related question. |
| Question 1. What risks do you see when executing commits directly into master instead of your branch?  Question 2. Why it is important to stay on your own branch; not to commit into branches of other teams?  Question 3. What risks can you envision, if the rebase of step (6) is skipped?  Question 4. What potential issues can result in a rebase? |
|  |
| Question 5.  **Follow the tutorial** for an introduction to GitHub. https://guides.github.com/activities/hello-world/  First, create your personal account at GitHub. Then, make sure your facilitator is invited to the repo  through these commands, <Settings>, <Manage Access>, <Invite Collaborator>.  @elentukh. @JMH JoshuaMichaelHanson.  For you to receive a full grade, your instructor needs to confirm these activities,   * Create a Repo * Create a Branch * Make a Commit * Open a Pull Request * Merge Pull Request   Question 6.  Students of the previous class developed the Git Repo Analysis Tool recommended for a team project. As an introduction to this tool, please access it at Firefox https://gitrepoanalysis.onrender.com/, then enter the URL with your personal repo and attach the resulting report to the submission of this assignment. You are encouraged to run the same report for your final presentation of team project.  Additional Notes   * People interpret the notion of 'collaboration' quite differently. It would not be a surprise to read about 'holding hands and singing cheerful tunes' as a 'collaboration'. In a context of this class, we delve into such technical tasks as 'rebase' and 'merge' as key parts of 'collaboration'. In GitHub, members of a development team, respond to Pull Requests with comments and approval of a code change. In module 5, when covering Continuous Delivery, most 'collaboration' steps are subsumed by an extensive regression test. * *Rebase* is an important concept in source control. During second module, we talk about *rebase* that can mess up your revision history. Here is the link to Git online book.   <https://git-scm.com/book/en/v2/Git-Branching-Rebasing>  This is a truly comprehensive collections of all kinds scenarios. Note that in other source  control systems, e.g. Clear Case, commands are different, although concepts remain the same.   * In this class, we prefer relying on a commonly used standard or a tradition. Brian Kernighan   starts his classic 1978 book titled "C - Programming Language" with a tutorial - how to printf "Hello World". Here we are a half a century later. See below the screenshot from GitHub tutorial, estimated as 10 minutes read. |

