Name: Analiza Amparo

**TEST I:**

1. c. git clone

2. c. git push

3. b. git status

4. c. git add

5. a. by committing the staged changes with git commit

6. c. git add

7. d. git commit –m “I’m coding”

8. d. git commit –am “I’m coding”

9. B. git push

10. B. git init

11. A. git switch <branch-name>

12. A. git remote –v

**TEST II:** Arrange the following basic git commands in the correct order.

1. Pushing changes:

A. git push

C. git commit -m “added home button”

B. git add homepage.php

2. Adding an existing project to GitHub.

C. git init

B. git add

A. git commit -m "first commit"

D. git remote add origin < your repository url >

E. git push –u origin master

**TEST III: Essay**

**1. What is Git version control?**

Git is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

**2. What are the benefits of Version Control System?**

Primary benefits you should expect from version control are as follows:

1. A complete long-term change history of every file. This means every change made by many individuals over the years. Changes include the creation and deletion of files as well as edits to their contents. Different VCS tools differ on how well they handle renaming and moving of files. This history should also include the author, date and written notes on the purpose of each change. Having the complete history enables going back to previous versions to help in root cause analysis for bugs and it is crucial when needing to fix problems in older versions of software. If the software is being actively worked on, almost everything can be considered an "older version" of the software.

2. Branching and merging. Having team members work concurrently is a no-brainer, but even individuals working on their own can benefit from the ability to work on independent streams of changes. Creating a "branch" in VCS tools keeps multiple streams of work independent from each other while also providing the facility to merge that work back together, enabling developers to verify that the changes on each branch do not conflict. Many software teams adopt a practice of branching for each feature or perhaps branching for each release, or both. There are many different workflows that teams can choose from when they decide how to make use of branching and merging facilities in VCS.

3. Traceability. Being able to trace each change made to the software and connect it to project management and bug tracking software such as Jira, and being able to annotate each change with a message describing the purpose and intent of the change can help not only with root cause analysis and other forensics. Having the annotated history of the code at your fingertips when you are reading the code, trying to understand what it is doing and why it is so designed can enable developers to make correct and harmonious changes that are in accord with the intended long-term design of the system. This can be especially important for working effectively with legacy code and is crucial in enabling developers to estimate future work with any accuracy.

**3. What is a Distributed System?**

Distributed computing is a field of computer science that studies distributed systems. A distributed system is a system whose components are located on different networked computers, which communicate and coordinate their actions by passing messages to one another.

**4. What are the differences between Git and GitHub?**

Git is a version control system that lets you manage and keep track of your source code history while GitHub is a cloud-based hosting service that lets you manage Git repositories. If you have open-source projects that use Git, then GitHub is designed to help you better manage them.