Poznan University of Technology
Object Oriented Programming
Version Control Systems

Jakub Piotr Hamerliński, M.Eng.

https://www.linkedin.com/in/hamerlinski

https://github.com/hamerlinski

Version Control Systems

Agenda

01. Git
02. GitHub
03. GitHub CLI

> *"Everything I have I owe to this job... this stupid, wonderful, boring, amazing job."* Jim Halpert

Version Control Systems

Git

Git is a version control system that allows you to track changes to your code over time and collaborate with other developers. It helps you keep track of all the changes you make to your code, as well as the changes made by your collaborators.

Version Control Systems

Git basic operations

There are a few basic operations in Git that you should be familiar with:

01. `git clone`: This command is used to create a local copy of a repository that is hosted on a remote server, such as GitHub.

02. `git add`: This command is used to add changes that you have made to your local repository to the staging area. This means that the changes are ready to be committed, but have not yet been permanently saved to the repository.

Version Control Systems

Git basic operations

03. `git commit`: This command is used to save changes that have been added to the staging area to the repository. When you commit changes, you should provide a message explaining what changes you made and why.
04. `git push`: This command is used to send your committed changes to a remote repository, such as GitHub.

Version Control Systems

Git basic operations

These are just a few of the basic operations in Git, but there are many more commands and features available. It’s a good idea to familiarize yourself with the basics of Git before diving into more advanced features and workflows

Version Control Systems

GitHub

GitHub is a web-based platform that is used for version control and collaboration on software projects. It allows developers to work on code together, track changes to the codebase, and roll back changes if necessary. It also provides a platform for users to host and share their software projects with others.

Git is the version control system that GitHub is built on top of. Git allows developers to track changes to a codebase over time, revert back to previous versions of the code, and collaborate with other developers on the same codebase.

Version Control Systems

GitHub

GitHub provides a user-friendly interface for developers to use Git, as well as a range of tools and features to make collaboration and project management easier.

GitHub is widely used by developers and organizations around the world to host and collaborate on software projects. It is also a popular resource for finding and sharing open-source software projects.

Version Control Systems

Example GitHub task

01. Create a new repository on GitHub.
02. Clone the repository to your local machine using the `git clone` command.
03. Create a new file in the repository and add some content to it.
04. Use `git add` and `git commit` to add the file to the repository and commit your changes.
05. Push the changes to the remote repository using `git push`.

Version Control Systems

Example GitHub task

01. Create a new branch in the repository using `git branch`.
02. Checkout the new branch using `git checkout`.
03. Make some changes to a file in the new branch and commit the changes.
04. Push the changes to the remote repository using `git push`.
05. Create a pull request to merge the changes from the new branch into the main branch.

Version Control Systems

GitHub CLI

GitHub CLI is a command line interface (CLI) tool that allows users to interact with the GitHub service and perform various actions directly from their terminal or command prompt. With GitHub CLI, users can create, clone, and fork repositories, create and edit issues and pull requests, and manage their git repositories and branches.

Version Control Systems

GitHub CLI

It also provides additional functionality such as the ability to search for repositories and users, and to manage and configure various aspects of their GitHub account. GitHub CLI is designed to be a faster and more efficient way to work with GitHub, and can be used in conjunction with other Git tools or as a standalone tool.

Version Control Systems

GitHub CLI installation

To install GitHub CLI, you will need to have a recent version of Git and Node.js installed on your system. Once these dependencies are installed, you can follow these steps:

01. Open your terminal or command prompt and enter the following command `npm install -g gh` or download installation file. This will install the GitHub CLI tool globally on your system.

Version Control Systems

GitHub CLI installation

02. Once the installation is complete, you can verify that the GitHub CLI is installed and available by running the following command `gh --version` This should print the version number of the GitHub CLI to your terminal.

Version Control Systems

GitHub CLI installation

03. If you want to configure the GitHub CLI with your GitHub account, you can run the following command `gh auth login`. This will open a web browser and prompt you to log in to your GitHub account. Once you have logged in, you will be asked to authorize the GitHub CLI to access your account.

Version Control Systems

GitHub CLI installation

04. Finally, you can test the GitHub CLI by running a simple command such as `gh repo list`. This should list all of the repositories in your GitHub account.

Version Control Systems

Example GitHub CLI task

01. Install GitHub CLI (if not installed):

    For macOS: `brew install gh`

    For Windows: `scoop install gh`

    For Linux:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-key C99B11DEB97541F0
sudo apt-add-repository https://cli.github.com/packages
sudo apt update
sudo apt install gh
```

02. Authenticate with your GitHub account: `gh auth login`
03. Clone a repository:

```
gh repo clone <repository-owner>/<repository-name>
cd <repository-name>
```

Version Control Systems

Example GitHub CLI task

4. Create a new branch: `git checkout -b <branch-name>`
5. Make changes to the files in the repository using your favorite text editor or IDE.
6. Add the changes and commit them:

```
git add .
git commit -m "feat: add new feature and didn't break code"
```

7. Push the branch to the remote repository: `git push --set-upstream origin <branch-name>`
8. Create a pull request:

```
gh pr create --title "Your PR title" --body "Description of the changes"
```

Version Control Systems

Example GitHub CLI task

09. Your pull request is now created on GitHub. You can view it by visiting the repository's pull request page on GitHub or by running the following command:

```
gh pr view --web
```

Remember to replace `<repository-owner>`, `<repository-name>`, `<branch-name>`, and other placeholders with appropriate values.

Thank you

Feel free to reach me via Linkedin

Fin