Object Oriented programming and software engineering – Lab 15

Adam Korytowski – 2025

I Version control systems.

Version Control Systems (VCS) are tools that help developers track changes in their code over time.
They allow you to:

- Save different versions of a project.
- Go back to an earlier version if something goes wrong.
- Collaborate with others, even when multiple people work on the same files.
- See who made which changes and when.

Think of VCS like a "time machine" for your code.

Without VCS, keeping backups manually (like copying folders) becomes messy and error-prone, especially on larger projects.

Why use Git specifically?

Feature	Explanation
Speed	Git is extremely fast because it runs locally (your computer has a full copy of the history).
Distributed System	Everyone has a full backup — not just a central server. You can work offline and sync changes later.
Branching and Merging	You can create "branches" to experiment or work on features separately without breaking the main project.
Strong Community Support	Tons of tutorials, tools (like GitHub), and integrations with popular IDEs (e.g., VSCode, IntelliJ).
Reliability	Git ensures data integrity; your project history is protected against corruption.
Free and Open Source	No cost, and you can study or contribute to Git itself if you wish.

II Git – configuration and file upload (6 pts)

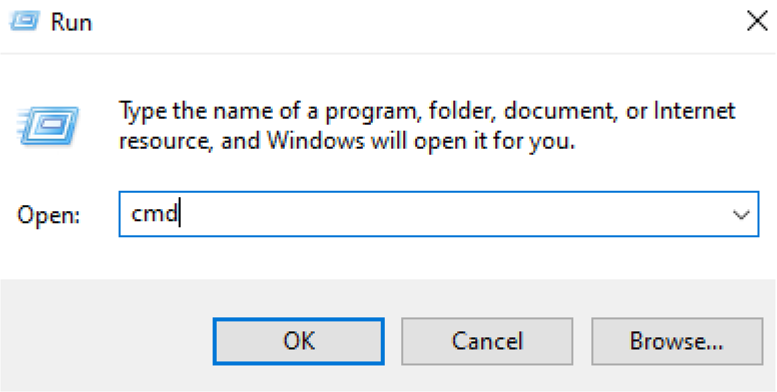
1. Create a GitHub Account using your @agh email account

- Go to: https://github.com/
- Click Sign up.
- Enter your @agh domain email address, username, password, and follow the instructions (you can skip adding a credit card — choose the free plan).
- Confirm your email address by clicking the link in the message sent by GitHub (if needed).

2. Install Git (and Git Bash) via Command Line

On Windows:

- Open Command Prompt



Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd

- Install Git using winget (Windows Package Manager). Run command:

```
winget install --id Git.Git -e --source winget
```

- Wait for installation to complete

Check installation:

- After installation, open Git-Bash and type:

```
git --version
```

- You should see something like git version 2.xx.x.

III Configure Git on Your Computer

- Set up your username and email:

```
git config --global user.name "Your Name"
git config --global user.email your@email.com
```

- Check your Git configuration:

```
git config --list
```

IV Create a New Repository on GitHub

- Log in to your GitHub account.
- Click the + icon in the top right corner → New repository.
- Enter a repository name (e.g., "Lab15"), optionally add a description.
- Choose Public or Private visibility.
- Important: Do NOT check the option "Initialize this repository with a README" (this helps when connecting an existing local repo).
- Click Create repository.

V Prepare Your Local Project Folder

- In Git Bash type:

```
cd <path to your code>
```

- Alternatively right-click in folder with your code, then press “Open Git Bash here”

VI Initialize a Local Git Repository

- In your project folder (still in Git Bash), run:

```
git init
```

7. Add Files to the Repository

- To add all files:

```
git add .
```

8. Make the First Commit

- Save your changes:

```
git commit -m "Initial commit"
```

9. Connect the Local Repository to GitHub

- Copy the repository URL from GitHub (e.g., https://github.com/your-username/Lab15.git).
- In Git Bash, run:

```
git remote add origin https://github.com/your-username/Lab15.git
```

10. Push Your Code to GitHub

- Push the code to the GitHub server:

```
git branch -M main
git push -u origin main
```

Verify if your changes are pushed into the repository:

- Open your account page on Github
- Find and click on your repository
- Observe if the files are uploaded