

GitHub Crash Course – Student Handout

1 Introduction to GitHub

- GitHub = Online platform for hosting and managing code/projects.
- Combines **Git (version control)** + **collaboration tools**.
- Used by developers, companies, and open-source communities.

💡 Why it matters:

- Build a **digital portfolio**.
- Show your projects to recruiters.
- Work in teams like real industry practice.

2 Understanding Git vs GitHub

- **Git** → A tool for tracking changes in files (like project history).
- **GitHub** → A website where you store Git projects online.
- Think of:
 - *Git = your personal diary*
 - *GitHub = publishing diary online so others can read & contribute*

3 Getting Started

- Create a **GitHub account** → <https://github.com>
- Install **Git** → <https://git-scm.com>
- Set up username & email:

```
git config --global user.name "Your Name"
```

```
git config --global user.email "youremail@example.com"
```

4 Your First Repository

- Login to GitHub → Click **New Repository**.
- Name: *portfolio-project*
- Add a **README.md** file → This explains your project.
- Clone to your computer:

```
git clone https://github.com/username/portfolio-project.git
```
- Add a file & push:

```
git add file.txt
```

```
git commit -m "First commit"
```

git push origin main

5 GitHub Workflow

- **Clone** = copy project
- **Commit** = save changes
- **Push** = upload to GitHub
- **Pull** = download latest changes

✦ Example workflow:

git pull

git checkout -b feature-1

git add .

git commit -m "Added new feature"

git push origin feature-1

6 Collaboration Tools

- **Branching** → Work without disturbing main code.
- **Pull Request (PR)** → Ask to merge your work.
- **Issues** → Track bugs or tasks.
- **Code Review** → Teammates check your code before merging.

7 Advanced Features

- **GitHub Pages** → Host portfolio websites free.
- **GitHub Actions** → Automate testing/deployment.
- **Projects & Wiki** → Manage tasks and documentation.

8 Building Your Digital Portfolio

1. Create repos for:
 - Mini projects
 - Final year project
 - Personal experiments
2. Add clear **README.md** files (description + screenshots).
3. Use **GitHub Pages** for hosting your personal website.
4. Share your GitHub profile link in resume/LinkedIn.

9 Best Practices

- Commit often with clear messages.

- Use meaningful repo names.
- Write documentation (README, comments).
- Keep code neat & structured.
- Contribute to open-source when possible.

10 Resources

- GitHub Docs → <https://docs.github.com>
- FreeCodeCamp Git/GitHub tutorials
- Learn Git Branching (game) → <https://learngitbranching.js.org>

✓ Session Wrap-Up

- GitHub = **Version control + Portfolio + Collaboration**
- Start small → push personal projects
- Build digital presence → recruiters will notice!

🧑 Next Step for You:

- 👉 Practice daily by uploading assignments or mini-projects to GitHub.