Course Introduction & Git/GitHub Setup

Due: 08/30/2025 *at 11:59 PM*
Points: 100

Why this assignment?

We’ll use Git and GitHub all term for submitting code, collaborating, and tracking progress. This is to make sure everyone:

- Has a working GitHub account with 2-factor authentication (2FA)

- Can use Git from their computer (Windows/macOS/Linux)

- Can clone, commit, push, branch, and open a Pull Request (PR)

- Follows our repository standards (README, license, .gitignore)

Learning objectives

By the end, you will be able to:

1. Create and secure a GitHub account with 2FA.

2. Install Git and configure your identity.

3. Authenticate to GitHub via SSH (preferred) or HTTPS.

4. Create a repo with a README and .gitignore; make commits and push.

5. Create a feature branch and open a Pull Request.

6. Use Issues and project hygiene basics (labels, descriptions, checklists).

What to submit

Post the following in the Brightspace submission box **and** in your repo's README.

Evidence:

1. **Repository URL** (public or invited access to instructor/TA).
2. **Pull Request URL** from your feature branch back to `main`.
3. **Issue URL** you created for this assignment.
4. A short reflection (3–5 sentences): what worked, what didn't, and one Git/GitHub concept you want to learn more about.

Part A — Accounts & security

1. **Create/verify GitHub account:** https://github.com
2. **Enable 2FA (required):** Settings → Password and authentication → Two-factor authentication → choose app-based (e.g., Authy, 1Password, Microsoft/Google Authenticator).
3. **Set your profile:** Add a professional name, avatar, bio (1–2 lines), and school/company.
4. **Student developer pack (optional):** https://education.github.com/pack

Deliverable A: In your README, include a checkbox list and check off each completed item.

```
## Assignment Evidence
- [x] GitHub account active
- [x] 2FA enabled (app-based)
- [x] Profile updated (name, avatar, short bio)
- [x] (Optional) Student Developer Pack applied / not eligible
```

Part B — Install & configure Git

Choose your OS and follow the steps. Use the terminal/PowerShell.

Windows
- Install: https://git-scm.com/download/win
- Use **Git Bash** or **PowerShell**.

macOS
- xcode-select --install (installs Command Line Tools) or install from https://git-scm.com/download/mac.

Linux (Debian/Ubuntu)
- sudo apt update && sudo apt install -y git

Configure your identity (all OS):

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

Deliverable B: Paste the output of git --version and your git config --global --list (redact personal email if desired) into your README

Part C — Authenticate with GitHub via SSH

1. **Generate SSH key** (Ed25519):

```
ssh-keygen -t ed25519 -C "your.email@example.com"
# Press Enter to accept default file; set a passphrase when prompted
```

2. **Start ssh-agent and add key**

- **Windows (PowerShell):**

```
Get-Service ssh-agent | Set-Service -StartupType Automatic
Start-Service ssh-agent
ssh-add $env:USERPROFILE\.ssh\id_ed25519
```

- **macOS/Linux:**

```
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_ed25519
```

3. **Copy your public key** and add to GitHub: Settings → SSH and GPG keys → New SSH key.

```
cat ~/.ssh/id_ed25519.pub
```

4. **Test connection:**

```
ssh -T git@github.com
# Expect: "Hi <username>! You've successfully authenticated..."
```

Alternative (HTTPS + token): create a **fine-grained personal access token** and use it as the password when prompted. *(OK, but SSH is recommended.)*

Deliverable C: Screenshot or copy the success line from `ssh -T` into your README (you may truncate your username).

Part D — Create repo, commit, push, branch, PR

1. **Create a new repo** named `cs-assign0-github-setup` (public or private with access granted). Initialize **with a README** and choose a language-appropriate **.gitignore**. Add an **MIT License**.
2. **Clone the repo:**

```
git clone git@github.com:<your-username>/cs-assign0-github-setup.git
cd cs-assign0-github-setup
```

3. **Create a feature branch:**

```
git checkout -b add-intro
```

4. **Edit README.md:** Add a section titled `About Me` (2–4 sentences) and paste your **Assignment 0 Evidence** checklists/outputs.
5. **Commit and push:**

```
git add README.md
git commit -m "Add About Me and Assignment evidence"
git push -u origin add-intro
```

6. **Open a Pull Request** from `add-intro` → `main`. Add a meaningful title/description.
7. **Create an Issue** titled “test issue” with a checklist of the parts A–D. Link the PR to the Issue.

Deliverable D: URLs for repo, PR, and Issue + updated README.

Part E — Reflection

Add a section at the end of your README titled `Reflection` and answer: - What went smoothly? - What was confusing? - One Git/GitHub concept you want to learn more about.

Troubleshooting FAQ

Q: ssh -T git@github.com says permission denied?
A: Ensure your public key is added in GitHub → SSH keys; confirm you added the **.pub** file; rerun ssh-add and verify the agent is running.

Q: Git asks for a username/password on push?
A: You're using HTTPS. Either switch remote to SSH:

```
git remote set-url origin git@github.com:<user>/<repo>.git
```

…or use a personal access token instead of your password.

Q: My commits show the wrong name/email.
A: Run git config --global user.name and git config --global user.email and recommit. Consider adding a **noreply** email from GitHub for privacy.

Q: Line endings on Windows look weird.
A: Set git config --global core.autocrlf true (Windows) or input (macOS/Linux).

Q: I can't push because of protected branches.
A: Open a PR from your feature branch; do not push directly to main.

Accessibility & alternatives

- If you cannot install software on your device, you may use **GitHub Codespaces** or **Dev Containers** (post a note in your README).