📌 GitHub Cheat Sheets→ FOR FACULTY

📝 Introduction: Key Terms Explained

Git – A tool that keeps track of changes in files and allows multiple people to work on the same project without confusion.
GitHub – A website that makes Git easy to use by storing projects online and letting people collaborate.
Repo (Repository) – A folder that holds all project files and their history. Like a shared Google Drive for code.
Clone – Making a personal copy of a GitHub project on your computer.
Branch – A separate copy of the project where changes can be made without affecting the main version.
Commit – Saving a set of changes to the project, like a "checkpoint" in a video game.
Push – Sending your saved changes (commits) from your computer to GitHub.
Pull Request (PR) – A request to add your changes to the main project. Others can review before approving.
Review of Pull Request – Checking someone's work before it is officially added to the project.
Merge – Combining changes from different branches into the main project.
Merge Conflict – When GitHub doesn't know which version of a file to keep because two people changed the same thing.
Latest Version – The most up-to-date version of the project on GitHub.
Undo Commit – Removing or fixing a saved change if you made a mistake.
Issue – A task, bug report, or discussion about improving the project.
Issue Management – Keeping track of tasks and progress using GitHub's built-in tools.

🎯 Quick Command Reference

Action	Command
Clone a repo	git clone URL
Create a new branch	git checkout -b branch-name
Switch to main branch	git checkout main
Update local project	git pull origin main
Save changes	git add . → git commit -m "message"
Upload changes	git push origin branch-name
Undo last commit	git reset --soft HEAD~1
Undo after pushing	git revert HEAD
Fix merge conflicts	Manually edit → git add . → git commit -m "fix"

💡 Final Tips

✅ Always **create a new branch** before working.
✅ **Pull latest updates** before starting new work.
✅ **Commit often** so you don't lose progress.
✅ **Communicate with teammates** to avoid merge conflicts.
✅ If stuck, **ask for help on GitHub Issues or in class**.

👩‍🏫 Cheat Sheet for Faculty: Managing Student Collaboration

🛠️ Setting Up a GitHub Classroom (Optional)

1. Go to https://classroom.github.com.
2. Create a **Classroom** and invite students using a link.
3. Assign projects and automatically generate student repositories.

👩‍🏫 Setting Up a GitHub Organization (Manual)

1. Create an organization: **GitHub → Profile → Your Organizations → New Organization**.
2. Create a **new repository** for the class.
3. Invite students via **Settings → Manage Access**.
4. Set up **branch protection** in **Settings → Branches** to prevent accidental overwrites.

👀 Reviewing Student Work

1. Go to the repository on **GitHub**.
2. Click **Pull Requests** → Find the student's request.
3. Review changes and leave comments if needed.
4. Click **"Merge"** if everything looks good.
5. Optionally, delete old branches to keep the project organized.

⚠️ Handling Merge Conflicts

If two students edit the same file, you may need to resolve conflicts:

1. Click the pull request with the conflict.
2. Follow GitHub's guided steps to choose the correct version.
3. Merge once resolved.

📌 GitHub Cheat Sheets→ FOR STUDENTS

📝 Introduction: Key Terms Explained

Git – A tool that keeps track of changes in files and allows multiple people to work on the same project without confusion.
GitHub – A website that makes Git easy to use by storing projects online and letting people collaborate.
Repo (Repository) – A folder that holds all project files and their history. Like a shared Google Drive for code.
Clone – Making a personal copy of a GitHub project on your computer.
Branch – A separate copy of the project where changes can be made without affecting the main version.
Commit – Saving a set of changes to the project, like a "checkpoint" in a video game.
Push – Sending your saved changes (commits) from your computer to GitHub.
Pull Request (PR) – A request to add your changes to the main project. Others can review before approving.
Review of Pull Request – Checking someone's work before it is officially added to the project.
Merge – Combining changes from different branches into the main project.
Merge Conflict – When GitHub doesn't know which version of a file to keep because two people changed the same thing.
Latest Version – The most up-to-date version of the project on GitHub.
Undo Commit – Removing or fixing a saved change if you made a mistake.
Issue – A task, bug report, or discussion about improving the project.
Issue Management – Keeping track of tasks and progress using GitHub's built-in tools.

🎯 Quick Command Reference

Action	Command
Clone a repo	git clone URL
Create a new branch	git checkout -b branch-name
Switch to main branch	git checkout main
Update local project	git pull origin main
Save changes	git add . → git commit -m "message"
Upload changes	git push origin branch-name
Undo last commit	git reset --soft HEAD~1
Undo after pushing	git revert HEAD
Fix merge conflicts	Manually edit → git add . → git commit -m "fix"

💡 Final Tips

✅ Always **create a new branch** before working.
✅ **Pull latest updates** before starting new work.
✅ **Commit often** so you don't lose progress.
✅ **Communicate with teammates** to avoid merge conflicts.
✅ If stuck, **ask for help on GitHub Issues or in class**.

🦁 Cheat Sheet for Students: Working on GitHub

📥 Getting Started: Cloning a Repository

1. Open **GitHub** and find your project.
2. Click **“Code” → Copy the HTTPS link**.
3. Open **Terminal / Command Prompt**, type:
4. git clone PASTE-LINK-HERE
5. Move into the folder:
6. cd PROJECT-NAME

✍️ Making Changes & Submitting Work

1. **Get the latest version before starting**:
2. git pull origin main
3. **Create a new branch for your work**:
4. git checkout -b my-feature
5. **Make changes** to files.
6. **Save your changes**:
7. git add .
8. git commit -m "Describe your changes here"
9. **Upload your work to GitHub**:
10. git push origin my-feature
11. **Create a Pull Request**:
    - Go to the repo on **GitHub.com**.
    - Click **Pull Requests → New Pull Request**.
    - Select **your branch**.
    - Click **"Create Pull Request"** and describe what you did.

✅ Updating Your Copy Before Working Again

1. **Switch to the main branch**:
2. git checkout main
3. **Get the latest updates**:
4. git pull origin main
5. **Create a new branch** and start working.

🔄 Fixing Mistakes

- **Undo last commit (before pushing):**
- git reset --soft HEAD~1
- **Undo last commit (after pushing):**
- git revert HEAD
- **Discard all uncommitted changes:**
- git checkout .

⚔️ Handling Merge Conflicts

If GitHub says there’s a conflict:

1. **Open the file** causing the conflict.
2. Look for lines marked with <<<<<<<, =======, >>>>>>>.
3. **Keep the correct version**, delete the conflict markers.
4. Save the file, then run:
5. git add .
6. git commit -m "Resolved merge conflict"
7. git push origin my-feature

📝 Using Issues for Tasks

- **To report a problem or track progress**, go to **Issues → New Issue**.
- Assign yourself an issue to work on.
- Close issues when done.