Git & GitHub Notes

Basic Concepts
- Git is a Version Control System that tracks changes in files
- Repository (Repo): A folder tracked by Git that contains your project files
- The repository contains a hidden `.git` folder with configuration, objects, and other Git-related information

Basic Git Workflow
1. Working Directory: Where you make changes to files
2. Staging Area: Files ready to be committed
3. Repository: Where Git stores the commit history

Essential Git Commands

Setup and Configuration
```bash
# Initialize a new Git repository
git init

# Configure Git globally
git config --global user.name "Your Name"
git config --global user.editor "preferred_editor"
```

Basic Operations
```bash
# Check repository status
git status

# Add files to staging area
```

```
git add filename.txt
git add . # Add all files

# Commit changes
git commit -m "commit message"

# View commit history
git log --oneline # Condensed view
```

Branches

```bash
# Create a new branch
git branch branch-name
git checkout -b branch-name # Create and switch to new branch

# Switch branches
git checkout branch-name
git switch branch-name # Modern alternative

# List branches
git branch # Local branches
git branch -r # Remote branches

# Merge branches
git merge branch-name
```

Working with Remote Repositories

```bash
# Add remote repository
git remote add origin repository-url

# Push changes
git push origin branch-name

# Pull changes
git pull
git fetch # Fetch without merging
```

Best Practices

Commit Guidelines
- Follow "Atomic Commits" principle: one commit per logical change
- Keep commits focused on one feature, component, or fix
- Write commit messages in present tense
- Make commit messages descriptive and specific

Collaboration Workflow
1. Create a branch for new features/fixes
2. Make changes and commit regularly
3. Push changes to remote repository
4. Create Pull Request (PR) for review
5. Address feedback and iterate
6. Merge changes after approval

Advanced Git Commands

Stashing

```bash
# Save changes temporarily
git stash

# Apply stashed changes
git stash apply
git stash pop # Apply and remove from stash
```

Comparing Changes
```bash
# View differences
git diff # Between working directory and staging
git diff branch1 branch2 # Between branches
```

Rebasing
- Alternative to merging
- Creates linear history
```bash
git rebase main
```

GitHub Specific

SSH Setup
- Generate SSH key to connect to GitHub
- Add SSH key to GitHub account settings

Contributing to Open Source
1. Fork the repository

2. Create feature branch
3. Make changes
4. Open Pull Request
5. Be patient and responsive to feedback
6. Follow project contribution guidelines

Best Practices
- Always pull before pushing new changes
- Keep local repository updated
- Use meaningful branch names
- Write clear PR descriptions
- Be responsive to review comments