

## অধ্যায়-১৯

### প্রজেক্ট এবং গিটহাব এডুকেশন

### (Project and GitHub Education)

#### ১৯.১ অধ্যায়ের উদ্দেশ্য

- এই অধ্যায় সমাপ্তির পর শিক্ষার্থীরা নিম্নলিখিত বিষয় জানতে পারবেঃ
- গিট এবং গিটহাব-এর অন্যান্য ব্যবহার
- গিটহাব-এ একাউন্ট এবং প্রজেক্ট তৈরি করা
- গিটহাব-এ প্রজেক্ট আপডেট / পুশ করা
- গিটহাব থেকে এক্সপ্লোর প্রজেক্ট ইমপোর্ট করা
- বহুল ব্যবহৃত কয়েকটি গিট কমান্ড।

#### ১৯.২ ভূমিকা

এই পুস্তকটি রচনার মূল উদ্দেশ্য হলো কমপিউটার প্রোগ্রামিং এবং অটোমেশন ইঞ্জিনিয়ারিং শেখা/চর্চার আগ্রহীদের হল বাস্তুব জীবনের ব্যবহার উপযোগী প্রোগ্রাম ডিজাইন এবং ডেভেলপ করার পথ-নির্দেশনা দেয়া। তবে বাস্তুব ভিত্তিক প্রোগ্রাম ডিজাইন, একটি পূর্ণাঙ্গ জাভা প্রজেক্ট তৈরির পদ্ধতি বর্ণনা, উপস্থাপন, বিশেষণ সহজ ব্যাপার নয়। এজন্য অনেক মেধার, সময়, সাধনা, ধৈর্য্য, ক্রিয়েটিভিটি এবং ব্রতের প্রয়োজন হয়। আর এরূপ একটি পূর্ণাঙ্গ প্রজেক্ট তৈরির পদ্ধতি বর্ণনা করতে সহস্রাধিক পাতার দামী পুস্তক রচনা/প্রকাশ করতে হয়। অনেক সময় এসব কাজে ব্যয়িত সময়ের তুলনায় অর্থ প্রাপ্তি খুবই সামান্য হতে পারে। এজন্য বিশেষ করে সময় এবং ধৈর্য্যের অভাবে অনেক মেধাবী, ক্রিয়েটিভ, সফল এবং নামকরা সফটওয়্যার ইঞ্জিনিয়ার পুস্তক লিখতে পারেন না। কিন্তু তারা যদি এগিয়ে আসতেন তবে দেশ-বিদেশে অবস্থানরত অনেক বাংলাদেশী ভাই-বোন সফল প্রোগ্রামার এবং অটোমেশন ইঞ্জিনিয়ার হতে পারতেন। এসব বিষয় বিবেচনা করে এবং এই পুস্তকের পৃষ্ঠাসংখ্যা এবং মূল্য সীমিত রাখার জন্য এই অধ্যায়ের প্রোগ্রাম এবং প্রজেক্টগুলো ইন্টারনেট / গিটহাবে ফ্রি প্রজেক্ট এবং ভিডিও আকারে প্রকাশ করার সিদ্ধান্ত নিয়েছি। এতে করে আগ্রহী পাঠকগণ ইন্টারনেট / গিটহাব থেকে শত-সহস্র প্রোগ্রাম/প্রজেক্ট ডাউনলোড করে প্রয়োজনীয় পরিবর্তন এবং পরিবর্তন করার মাধ্যমে দক্ষতা অর্জন করতে পারবেন। তবে এজন্য সামান্য দুটো বিষয় - গিট এবং গিটহাব এবং এদের ব্যবহারবিধি সম্পর্কে জানা দরকার। শূন্য অধ্যায়ে জাভা প্রজেক্ট তৈরি, প্রজেক্ট এক্সপোর্ট / ইমপোর্ট পদ্ধতি ইত্যাদি সম্পর্কে বিস্তারিত আলোচনা করা হয়েছে। এ অধ্যায়ে গিট এবং গিটহাব-এর অন্যান্য ব্যবহার; গিটহাব-এ একাউন্ট এবং প্রজেক্ট তৈরি করা; গিটহাব-এ প্রজেক্ট আপডেট / পুশ করা; গিটহাব থেকে এক্সপ্লোর প্রজেক্ট ইমপোর্ট করা এবং বহুল ব্যবহৃত কয়েকটি গিট কমান্ড সম্পর্কে সংক্ষেপে আলোচনা করা হয়েছে। গিট এবং গিটহাব দুটো মজার বিষয় যা কোম্পানি এবং প্রোগ্রামারদের কাছে খুবই জনপ্রিয় এবং বহুল ব্যবহৃত গুরুত্বপূর্ণ বিষয়।

#### ১৯.৩ গিট-একটি ভার্সন কন্ট্রোল সিস্টেম (Git- A Version Control System)

গিট (Git) হল কমপিউটারে ফাইল পরিবর্তনের ট্র্যাকিং সংরক্ষণের এবং একাধিক জনের সাথে কাজ করার জন্য একটি সহজ এবং জনপ্রিয় সংস্করণ নিয়ন্ত্রণ ব্যবস্থা (Version Control System)। এটি মূলত সফটওয়্যার ডেভেলপমেন্টের সময় সোর্স কোড সংরক্ষণ এবং পরিচালনার জন্য ব্যবহৃত হয়। এটি কোন ফাইল বা প্রজেক্টের ফাইলগুলোর পরিবর্তনের ট্র্যাক সংরক্ষণ করতে এবং নিয়ন্ত্রণ রাখতে পারে। গিট সম্পর্কে বিস্তারিত জানার জন্য ভিজিট করুনঃ <https://en.wikipedia.org/wiki/Git>

#### ১৯.৩.১ গিট ইনস্টলেশন (Git Installation)

গিট কমান্ডগুলো গিট-এর নিজস্ব বা ডস-এর কমান্ড উইন্ডোতে টাইপ করে কার্যকর করতে হয়। তবে এখন পর্যন্ত উইন্ডোজ এবং ম্যাক অপারেটিং সিস্টেমে গিট কমান্ডগুলো বিস্ট-ইন নয়। এজন্য গিট ওয়েব সাইট থেকে গিট ইনস্টলার ডাউনলোড করে ইনস্টল করে নিতে হয়। আপনার কমপিউটারে গিট ইনস্টল করার জন্য <https://git-scm.com/> ওয়েব সাইটের ডাউনলোড (Download) বাটনে / পাতায় ক্লিক করে আপনার কমপিউটারের অপারেটিং সিস্টেম এবং প্রসেসরের ভার্সন অনুযায়ী উপযুক্ত গিট ইনস্টলারটি ডাউনলোড করুন। যেমন, ৬৪-বিট উইন্ডোজ অপারেটিং সিস্টেমের জন্য Git-2.13.0-64-bit.exe নামক গিট ইনস্টলারটি ডাউনলোড করতে হবে। অতপর আপনার কমপিউটারের ডাউনলোড লোকেশন (ডিফল্ট ডাউনলোড লোকেশনঃ Downloads) গিয়ে তাতে ডাবল ক্লিক করে ইনস্টল করুন। ঠিকমত ইনস্টল হলে কমান্ড উইন্ডোতে git --version কমান্ড লিখে এন্টার কী চাপলে নিম্নরূপ গিট ভার্সন ইনফরমেশন দেখতে পাবেন। না পেলে কমপিউটার রিস্টার্ট করুন কিংবা পুনরায় গিট ইনস্টল করুন। উল্লেখ্য, প্রতিটি গিট কমান্ড git দিয়ে শুরু হয়।

```

Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\mhossa200>git --version
git version 2.13.0.windows.1

C:\Users\mhossa200>

```

মনে রাখুনঃ গিট কমান্ডগুলো কমান্ড উইন্ডোতে সম্পাদন করতে হয় এবং প্রতিটি গিট কমান্ড **git** দিয়ে শুরু হয়।

### ১৯.৩.২ গিট চিটশীট (Git Cheat Sheet):

গিট (Git) অনেকটা ডস কমান্ডের (DOS Command) মত ব্যবহৃত হয়। অর্থাৎ উইন্ডোজে কমান্ড উইন্ডো কিংবা ম্যাকে টার্মিনাল উইন্ডোতে ডস- কমান্ডের মত কমান্ড লিখে গিট অপারেশন সম্পাদন করতে হয়। মাত্র কয়েকটি (১৫-২০) গিট কমান্ড জানা থাকলে গিটহাব থেকে শত-সহস্র প্রোগ্রাম/প্রজেক্ট ডাউনলোড করে প্রয়োজনীয় পরিবর্তন এবং পরিবর্তন করার মাধ্যমে দক্ষতা অর্জন করতে পারবেন। নিম্নে এরূপ কতগুলো কমান্ড দেয়া হল। প্রথম দিকে কতগুলো গিট কমান্ড কঠিন এবং বিরজিকর মনে হলেও একবার এগুলো ব্যবহারের মজা বুঝতে পারলে আপনি নিশ্চয়ই খুশী এবং দক্ষ হবেন। বিস্তারিত জানার জন্য গিটহাব এডুকেশন (<https://education.github.com>) ওয়েব সাইটে প্রকাশিত বিভিন্ন টিউটোরিয়াল কিংবা এই পুস্তকের গিটহাব সাইটে (<https://github.com/icterguru/JavaProgrammingA2Z>) ওয়েব Useful\_Links.html ফাইলে প্রকাশিত বিভিন্ন ভিডিও দেখুন কিংবা ইন্টারনেটে / ইউটিউবে সার্চ করুন।

#### INSTALLATION & GUI

With platform specific installers for Git, GitHub also provides the ease of staying up-to-date with the latest releases of the command line tool while providing a graphical user interface for day-to-day interaction, review, and repository synchronization.

**GitHub for Windows**  
<https://windows.github.com>

**GitHub for Mac**  
<https://mac.github.com>

For Linux and Solaris platforms, the latest release is available on the official Git web site.

**Git for All Platforms**  
<http://git-scm.com>

#### SETUP

Configuring user information used across all local repositories

<b>git config --global user.name "[firstname lastname]"</b>
set a name that is identifiable for credit when review version history
<b>git config --global user.email "[valid-email]"</b>
set an email address that will be associated with each history marker
<b>git config --global color.ui auto</b>
set automatic command line coloring for Git for easy reviewing

#### SETUP & INIT

Configuring user information, initializing and cloning repositories

<b>git init</b>
initialize an existing directory as a Git repository
<b>git clone [url]</b>
retrieve an entire repository from a hosted location via URL

#### STAGE & SNAPSHOT

Working with snapshots and the Git staging area

<b>git status</b>
show modified files in working directory, staged for your next commit
<b>git add [file]</b>
add a file as it looks now to your next commit (stage)
<b>git reset [file]</b>
unstage a file while retaining the changes in working directory
<b>git diff</b>
diff of what is changed but not staged
<b>git diff --staged</b>
diff of what is staged but not yet committed
<b>git commit -m "[descriptive message]"</b>
commit your staged content as a new commit snapshot

#### BRANCH & MERGE

Isolating work in branches, changing context, and integrating changes

<b>git branch</b>
list your branches. a* will appear next to the currently active branch
<b>git branch [branch-name]</b>
create a new branch at the current commit
<b>git checkout</b>
switch to another branch and check it out into your working directory
<b>git merge [branch]</b>
merge the specified branch's history into the current one
<b>git log</b>
show all commits in the current branch's history

## INSPECT & COMPARE

Examining logs, diffs and object information

<b>git log</b>
show the commit history for the currently active branch
<b>git log branchB...branchA</b>
show the commits on branchA that are not on branchB
<b>git log --follow [file]</b>
show the commits that changed file, even across renames
<b>git diff branchB...branchA</b>
show the diff of what is in branchA that is not in branchB
<b>git show [SHA]</b>
show any object in Git in human-readable format

## SHARE & UPDATE

Retrieving updates from another repository and updating local repos

<b>git remote add [alias] [url]</b>
add a git URL as an alias
<b>git fetch [alias]</b>
fetch down all the branches from that Git remote
<b>git merge [alias]/[branch]</b>
merge a remote branch into your current branch to bring it up to date
<b>git push [alias] [branch]</b>
Transmit local branch commits to the remote repository branch
<b>git pull</b>
fetch and merge any commits from the tracking remote branch

## TRACKING PATH CHANGES

Versioning file removes and path changes

<b>git rm [file]</b>
delete the file from project and stage the removal for commit
<b>git mv [existing-path] [new-path]</b>
change an existing file path and stage the move
<b>git log --stat -M</b>
show all commit logs with indication of any paths that moved

## REWRITE HISTORY

Rewriting branches, updating commits and clearing history

<b>git rebase [branch]</b>
apply any commits of current branch ahead of specified one
<b>git reset --hard [commit]</b>
clear staging area, rewrite working tree from specified commit

## TEMPORARY COMMITS

Temporarily store modified, tracked files in order to change branches

<b>git stash</b>
Save modified and staged changes
<b>git stash list</b>
list stack-order of stashed file changes
<b>git stash pop</b>
write working from top of stash stack
<b>git stash drop</b>
discard the changes from top of stash stack

## IGNORING PATTERNS

Preventing unintentional staging or committing of files

<b>logs/ *.notes pattern*/</b>
Save a file with desired patterns as .gitignore with either direct string matches or wildcard globs.
<b>git config --global core.excludesfile [file]</b>
system wide ignore pattern for all local repositories

Source: <https://education.github.com/git-cheat-sheet-education.pdf>

## ১৯.৪ গিটহাব-একটি প্রজেক্ট ম্যানেজমেন্ট সিস্টেম (Git –A Project Management System)

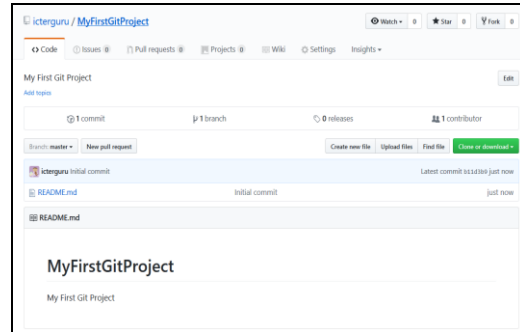
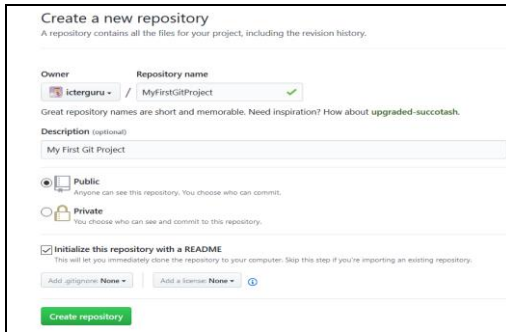
গিটহাব (GitHub) হল একটি ওয়েব ভিত্তিক গিট বা সংস্করণ/নিয়ন্ত্রণস্থল এবং ইন্টারনেট হোস্টিং পরিষেবা। এটি মূলত গিটের বিতরণকৃত সোর্স কোড সংস্করণ, নিয়ন্ত্রণ এবং সোর্স কোড ম্যানেজমেন্ট কার্যকারিতা এবং তার নিজস্ব বৈশিষ্ট্যাবলী যোগ করার সুযোগ প্রদান করে। এটি প্রতিটি প্রজেক্টের জন্য অ্যাক্সেস কন্ট্রোল এবং বিভিন্ন সহযোগিতামূলক বৈশিষ্ট্য যেমন বাগ ট্র্যাকিং, ফিচার অনুরোধ, টাঙ্ক ম্যানেজমেন্ট প্রদান করে। গিটহাবের একই একাউন্টে গোপনীয় এবং উন্মুক্ত (Private and Public) প্রজেক্টের সোর্স কোড সংরক্ষণ করা যায়। গিটহাবের প্রাইভেট প্রজেক্ট সংরক্ষণের জন্য মূল্য দিতে হয় কিন্তু পাবলিক প্রজেক্টের সংরক্ষণ এবং নিয়ন্ত্রণ ফ্রি। গিটহাবের পাবলিক প্রজেক্ট থেকে যে কেউ সোর্স কোড ডাউনলোড করে নিতে পারে এবং কর্পোরেট কোম্পানিগুলো তাদের নিজস্ব প্রজেক্ট সংরক্ষণ এবং নিয়ন্ত্রণের জন্য তাদের কর্মীদের জন্য প্রাইভেট একাউন্ট ক্রয় করে। গিটহাব বর্তমান বিশ্বের সবচেয়ে জনপ্রিয় এবং বৃহত্তম সোর্সকোড মাধ্যম হিসেবে পরিচিত। এপ্রিল ২০১৭ পর্যন্ত গিটহাব প্রায় ২০ মিলিয়ন ইউজারের ৫৭ মিলিয়ন প্রজেক্ট হোস্ট করেছেন। গিট এবং গিটহাব সম্পর্কে বিস্তারিত জানার জন্য <https://github.com/>, <https://education.github.com/> প্রভৃতি ওয়েব সাইট ভিজিট করুন।

### ১৯.৪.১ গিটহাবে একাউন্ট তৈরি (Creating an Account in GitHub)

গিটহাবে একাউন্ট তৈরি করার জন্য <https://github.com/> ওয়েব সাইটে প্রবেশ করুন। অতপর বর্ণিত ধাপগুলো অনুসরণ করুন। এজন্য আপনার একটি ই-মেইল একাউন্ট থাকতে হবে। যেখানে আপনার নতুন হিটহাব একাউন্টটি একটিভ করার জন্য কিংবা হিটহাব ইউজারনেম কিংবা পাসওয়ার্ড উদ্ধার করার জন্য গিটহাব থেকে আপনাকে স্বয়ংক্রিয়ভাবে ই-মেইল প্রেরিত হবে। একটি একটি সহজ বিষয় বিধায় এখানে বিস্তারিত আলোচনা করা হল না। এজন্য <https://github.com/>, <https://education.github.com/> প্রভৃতি ওয়েব সাইট ভিজিট করুন কিংবা এই পুস্তকের গিটহাব সাইটে (<https://github.com/icterguru/JavaProgrammingA2Z>) প্রদত্ত [Useful\\_Links.html](#) ফাইলে প্রকাশিত বিভিন্ন ভিডিও দেখুন কিংবা ইন্টারনেটে / ইউটিউবে সার্চ করুন।

### ১৯.৪.২ গিটহাবে প্রজেক্ট তৈরি (Creating Projects in GitHub)

গিটহাবে একটি নতুন প্রজেক্ট তৈরি করার জন্য (<https://github.com/new>) ওয়েব সাইটে প্রবেশ করুন। অতপর বর্ণিত ধাপগুলো অনুসরণ করুন। প্রয়োজনে আপনার হিটহাব একাউন্টে লগ-ইন করুন কিংবা নতুন একটি হিটহাব একাউন্ট তৈরি করুন। গিটহাবে স্বয়ংক্রিয়ভাবে কিংবা কমান্ড উইন্ডোতে কতগুলো গিট কমান্ড সম্পাদন করে নতুন প্রজেক্ট তৈরি করা যায়। নমুনা হিসেবে নিম্নের ছবিতে গিটহাবে MyFirstGitProject নামে একটি নতুন প্রজেক্ট বা রিপজিটরি (Repository) তৈরি করার পদ্ধতি দেখানো হয়েছে। আপনার একাউন্টে এরূপ একটি প্রজেক্ট তৈরির জন্য ছবছ নিচের পদ্ধতি অনুসরণ করুন।



উপরোক্ত MyFirstGitProject নামের প্রজেক্টটি গিটহাব রিপজিটরিতে তৈরি হয়েছে বটে তবে এটি সম্পূর্ণ প্রজেক্ট নয়। এতে উপরোক্ত README.md ব্যতীত অন্য কোন ফাইল নেই। যদিও এতে নতুন কোন ফাইল তৈরি করে কিংবা বিদ্যমান কোন ফাইল আপলোড করা যায়। কিন্তু একাধিক ফাইল এবং ফোল্ডার সমৃদ্ধ পূর্ণাঙ্গ কোন প্রজেক্ট সংযোজন সময় সাপেক্ষ। এজন্য অভিজ্ঞ প্রোগ্রামারগণ সাধারণত কমান্ড উইন্ডোতে কতগুলো গিট কমান্ড সম্পাদন করে নতুন প্রজেক্ট তৈরি করা যায়।

উল্খ্য, গিটহাবে স্বয়ংক্রিয়ভাবে কোন প্রজেক্ট তৈরির জন্য Initialize this repository with a README চেকবক্স সিলেক্ট করা আবশ্যিক। অন্যথায় কমান্ড উইন্ডোতে কতগুলো গিট কমান্ড সম্পাদন করে এই প্রজেক্টটি সম্পন্ন করার জন্য নিম্নের ধাপগুলো অনুসরণ করতে বলা হত।

#### create a new repository on the command line

```
echo "# MyFirstGitProject" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/icterguru/MyFirstGitProject.git
git push -u origin master
```

নিম্নে উপরোক্ত পদ্ধতিতে গিট কমান্ড সম্পাদন করে এই পুস্তকের বিভিন্ন প্রোগ্রাম এবং প্রজেক্ট সংরক্ষণের জন্য JavaProgrammingA2Z প্রজেক্টটি তৈরির ধাপগুলো দেয়া হল।

- 1) On Eclipse, Create a Maven Project as "JavaProgrammingA2Z"
- 2) C:\Users\mhossaa200\Workspace>cd JavaProgrammingA2Z
- 3) C:\Users\mhossaa200\Workspace\JavaProgrammingA2Z>git init  
Initialized empty Git repository in C:\Users\mhossaa200\Workspace\JavaProgrammingA2Z\.git\
- 4) C:\Users\mhossaa200\Workspace\JavaProgrammingA2Z>git add .
- 5) C:\Users\mhossaa200\Workspace\JavaProgrammingA2Z>git commit -m "first commit"  
[master (root-commit) 11db44b] first commit  
Committer: Guru <icterguru@gmail.com>  
Your name and email address were configured automatically based

on your username and hostname. Please check that they are accurate.  
You can suppress this message by setting them explicitly. Run the following command and follow the instructions in your editor to edit your configuration file:

```
git config --global --edit
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
13 files changed, 175 insertions(+)
create mode 100644 .classpath
create mode 100644 .project
create mode 100644 .settings/org.eclipse.core.resources.prefs
create mode 100644 .settings/org.eclipse.jdt.core.prefs
create mode 100644 .settings/org.eclipse.m2e.core.prefs
create mode 100644 pom.xml
create mode 100644 src/main/java/com/icterguru/JavaProgrammingA2Z_gid/com/icterguru/JavaProgrammingA2Z/App.java
create mode 100644 src/test/java/com/icterguru/JavaProgrammingA2Z_gid/com/icterguru/JavaProgrammingA2Z/AppTest.java
create mode 100644 target/classes/META-INF/MANIFEST.MF
create mode 100644 target/classes/META-INF/maven/com.icterguru.JavaProgrammingA2Z_gid/com.icterguru.AutomationJavaSeleniumAbc/pom.properties
create mode 100644 target/classes/META-INF/maven/com.icterguru.JavaProgrammingA2Z_gid/com.icterguru.AutomationJavaSeleniumAbc/pom.xml
create mode 100644 target/classes/com/icterguru/JavaProgrammingA2Z_gid/com/icterguru/JavaProgrammingA2Z/App.class
create mode 100644 target/test-classes/com/icterguru/JavaProgrammingA2Z_gid/com/icterguru/JavaProgrammingA2Z/AppTest.class
```

6) C:\Users\mhossain200\Workspace\JavaProgrammingA2Z>git remote add origin <https://github.com/icterguru/JavaProgrammingA2Z.git>

7) C:\Users\mhossain200\Workspace\JavaProgrammingA2Z>git push -u origin master

If found.....

remote: Permission to icterguru/JavaProgrammingA2Z.git denied to mokterhossain.  
fatal: unable to access '<https://github.com/icterguru/JavaProgrammingA2Z.git/>': The requested URL returned error: 403

From <https://github.com/icterguru/JavaProgrammingA2Z/settings>

Add mokterhossain as a contributor and ask the contributor to accept your invitation from his/her github account, following the automated email sent to his/her (mokterhossain) email

Then execute the above command

8) C:\Users\mhossain200\Workspace\JavaProgrammingA2Z>git push -u origin master

```
Counting objects: 51, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (30/30), done.
Writing objects: 100% (51/51), 5.12 KiB | 0 bytes/s, done.
Total 51 (delta 0), reused 0 (delta 0)
To https://github.com/icterguru/JavaProgrammingA2Z.git
 * [new branch] master -> master
Branch master set up to track remote branch master from origin.
```

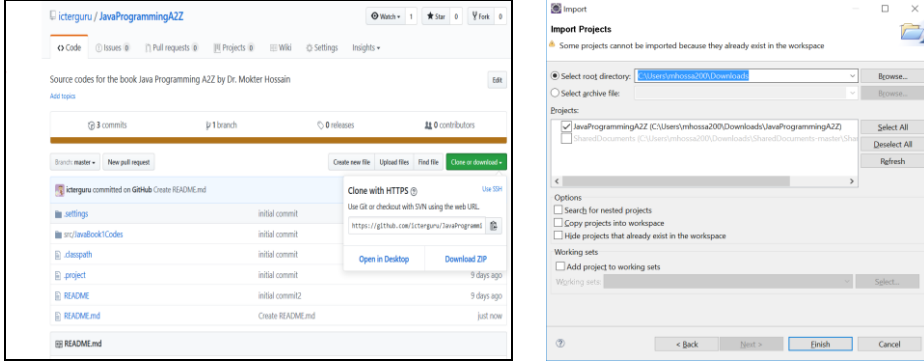
9) Now you should see the complete project uploaded in your Github and JavaProgrammingA2Z [JavaProgrammingA2Z master] in your Eclipse

10) If not, remove the pproject from Eclipse (DO NOT REMOVE from Disk) and Relimport it.

নতুনদের জন্য গিটহাবে একটি প্রজেক্ট তৈরি করা খুব একটা সহজ বিষয় নয় বিধায় এই পুস্তকের সীমিত পরিসরে এখানে বিস্তারিত বর্ণনা করা সম্ভব হলো না। এজন্য <https://github.com/>, <https://education.github.com/> প্রভৃতি ওয়েব সাইট ভিজিট করুন কিংবা এই পুস্তকের গিটহাব প্রজেক্টে প্রদত্ত Useful\_Links.html ফাইলে প্রকাশিত বিভিন্ন ভিডিও দেখুন কিংবা ইন্টারনেটে / ইউটিউবে সার্চ করুন।

### ১৯.৪.৩ গিটহাব থেকে প্রজেক্ট ডাউনলোড করা (Downloading Projects from GitHub)

গিটহাব থেকে কোন প্রজেক্ট ডাউনলোড করার জন্য প্রথমে যে কোন ওয়েব ব্রাউজারের এড্রেসবারে প্রজেক্টটির ওয়েব এড্রেস লিখে সেখানে প্রবেশ করুন। যেমন, এই পুস্তকের গিটহাব প্রজেক্টের এড্রেস <https://github.com/icterguru/JavaProgrammingA2Z> লিখলে আপনি নিম্নরূপ উইন্ডোতে প্রজেক্টের ফাইল এবং ফোল্ডারগুলো দেখতে পাবেন। অতঃপর Download ZIP বাটনে ক্লিক করলে এটি আপনার ডাউনলোড ফোল্ডারে JavaProgrammingA2Z-master.zip নামে একটি জিপ ফাইল হিসেবে সেভ হবে। সেটি আনজিপ করে -master অংশটি বাদ দিয়ে JavaProgrammingA2Z আকারে আপনার এক্সপ্লোরার ফোল্ডারে ইমপোর্ট করতে পারেন।



### ১৯.৪.৪ গিটহাব থেকে এক্সিলে প্রজেক্ট ইমপোর্ট করা (Importing a GitHub Project in Eclipse)

এক্সিলে উপরোক্ত প্রজেক্টটি ইমপোর্ট করার জন্য এক্সিলে-এর Project Explorer উইন্ডো থেকে File > Import... নির্বাচন করুন। শূন্য অধ্যায়ে এ সম্পর্কে আলোচনা করা হয়েছে। প্রয়োজনে এই পুস্তকের গিটহাব প্রজেক্টে প্রদত্ত Useful\_Links.html ফাইলে প্রকাশিত বিভিন্ন ভিডিও দেখুন কিংবা ইন্টারনেটে / ইউটিউবে সার্চ করুন।

### ১৯.৪.৫ গিটহাব থেকে কোন প্রজেক্ট ক্লোন করা (Cloning Projects from GitHub)

কমান্ড উইন্ডোতে গিট কমান্ড সম্পাদন করে উপরোক্ত **JavaProgrammingA2Z** প্রজেক্টটি ক্লোন করার জন্য যে ওয়ার্কস্পেস ফোল্ডারে প্রজেক্টটি ক্লোন করতে চান সেখানে প্রবেশ করুন। অতপর জাস্ট নিম্নের এক লাইনের কমান্ডটি সম্পাদন করুন। **git clone https://github.com/icterguru/JavaProgrammingA2Z.git** এখানে **JavaProgrammingA2Z.git** হল কাস্থিত প্রজেক্টটির নাম। লক্ষ্য করুন, গিটহাব প্রজেক্টটির Clone with HTTPS অপশনের ঠিক নিচে এর পূর্ণাঙ্গ এড্রেসটি দেয়া আছে। পার্শ্ববর্তী কপি আইকনে ক্লিক করে এটি কপি করে নিতে পারেন।

<https://github.com/icterguru/JavaProgrammi>

এভাবে ক্লোন করলে JavaProgrammingA2Z প্রজেক্টটি আপনার এক্সিলে ওয়ার্কস্পেস ফোল্ডারে সরাসরি কপি হবে। এটি আর আনজিপ করার প্রয়োজন হয় না বিধায় এটি সরাসরি এক্সিলে ওয়ার্কস্পেসে ইমপোর্ট করা যায়। উপরোক্ত টিমওয়ার্ক পদ্ধতিতে সময়ে সময়ে সর্বশেষ আপডেটেড প্রজেক্ট ডাউনলোড করাসহ অনেক কাজে ক্লোনিং পদ্ধতি সুবিধাজনক।

গিটহাবে উপরোক্ত কার্যাবলীসগণ আরও কতগুলো বিশেষ কাজ যেমন প্রজেক্টের নতুন কোন ব্রাঞ্চ তৈরি করা, প্রোগ্রাম কোড এডিট করে পুনরায় পুশ করা, বাগ ফিক্সিং, কনফ্লিক্ট ম্যানেজমেন্ট, ইত্যাদি। নতুনদের জন্য গিটহাবে এগুলো করা খুব একটা সহজ বিষয় নয় বিধায় এই পুস্তকের সীমিত পরিসরে এখানে বিস্তারিত বর্ণনা করা সম্ভব হলো না। এজন্য <https://github.com/>, <https://education.github.com/> ওয়েব সাইট ভিজিট করুন কিংবা এই পুস্তকের গিটহাব প্রজেক্টে প্রদত্ত Useful\_Links.html ফাইলে প্রকাশিত বিভিন্ন ভিডিও দেখুন কিংবা ইন্টারনেটে / ইউটিউবে সার্চ করুন।

### More about GitHub Education:

<https://education.github.com>  
<https://github.com/icterguru/JavaProgrammingA2Z>  
<https://github.com/icterguru/>

### ১৯.৫ পরিসমাপ্তি (Conclusion)

পরিশেষে বলতে চাই, এই পুস্তকের পাতার সংখ্যা এবং মূল্য সীমিত রাখার উদ্দেশ্য এই অধ্যায়ের বিষয়গুলো বিস্তারিত আলোচনা না করে ভিডিও আকারে প্রকাশ করা হবে। এটি পাঠকদের জন্য অধিকতর উপকারী হবে বলে আশা করা যায়। মনে রাখবেন, একটি চিত্র হাজারো বাক্য বর্ণনা করে এবং একটি ভিডিও শত-সহস্র স্থিরচিত্রের সমন্বয়ে গঠিত হয়। সুতরাং এই অধ্যায়ের বর্ণনা এবং ভিডিওগুলো প্রকাশিত হলে নবীন এবং অভিজ্ঞ সব ধরনের পাঠকদের জন্য উপকারী হবে বলে আশা করছি।