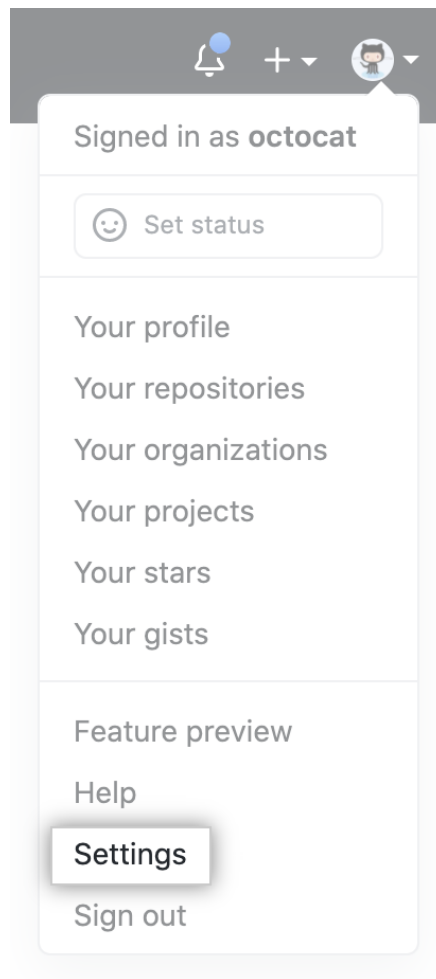


Classwork-1.0: Prepare for GitHub submission

GitHub requires Personal access tokens (PATs) for authentication in git operation. You can create a PAT as followed.

1. In the upper-right corner of any page in GitHub website, click your profile photo, then click Settings.



2. In the left sidebar, click Developer settings.

Organizations

Moderation

Code, planning, and automation

Repositories

Packages

GitHub Copilot

Pages

Saved replies

Security

Code security and analysis

Integrations

Applications

Scheduled reminders

Archives

Security log

Sponsorship log

<> Developer settings

You can @mention other users and organizations to link to them.

URL

Twitter username

Company

You can @mention your company's GitHub organization to link it.

Location

All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to share this data wherever your user profile appears. Please see our [privacy statement](#) to learn more about how we use this information.

Update profile

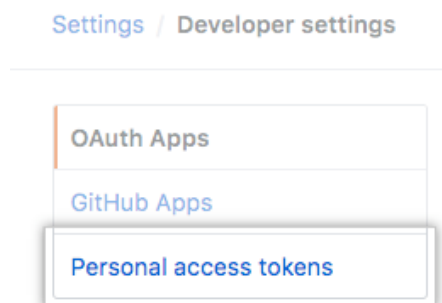
Contributions & Activity

☐ **Make profile private and hide activity** Beta
Enabling this will hide your contributions and activity from your GitHub profile and from social features like followers, stars, feeds, leaderboards and releases.

☐ **Include private contributions on my profile**
Your contribution graph, achievements, and activity overview will show your private contributions without revealing any repository or organization information. [Read more.](#)

Update preferences

3. In the left sidebar, click Personal access tokens.



4. Click Generate new token.
5. Give your token a descriptive name.
6. To give your token an expiration, select the Expiration drop-down menu, then click a default or use the calendar picker.

Expiration

7 days

The token will expire on Friday, Feb 8 2008

7. Select the scopes, or permissions, you'd like to grant this token. To use your token to access repositories from the command line, select repo.

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> admin:org	Full control of orgs and teams
<input type="checkbox"/> write:org	Read and write org and team membership
<input type="checkbox"/> read:org	Read org and team membership
<input type="checkbox"/> admin:public_key	Full control of user public keys
<input type="checkbox"/> write:public_key	Write user public keys
<input type="checkbox"/> read:public_key	Read user public keys
<input type="checkbox"/> admin:repo_hook	Full control of repository hooks
<input type="checkbox"/> write:repo_hook	Write repository hooks
<input type="checkbox"/> read:repo_hook	Read repository hooks
<input type="checkbox"/> admin:org_hook	Full control of organization hooks
<input type="checkbox"/> gist	Create gists
<input type="checkbox"/> notifications	Access notifications
<input type="checkbox"/> user	Update all user data
<input type="checkbox"/> user:email	Access user email addresses (read-only)
<input type="checkbox"/> user:follow	Follow and unfollow users
<input type="checkbox"/> delete_repo	Delete repositories

8. Click Generate token.

9. Store the PAT in a safety location.

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your new personal access token now. You won't be able to see it again!

✓ ghp_IqIMN0ZH6z0wIEB4T9A2g4EHMy8Ji42q4HA5 

Enable SSO ▾

Delete

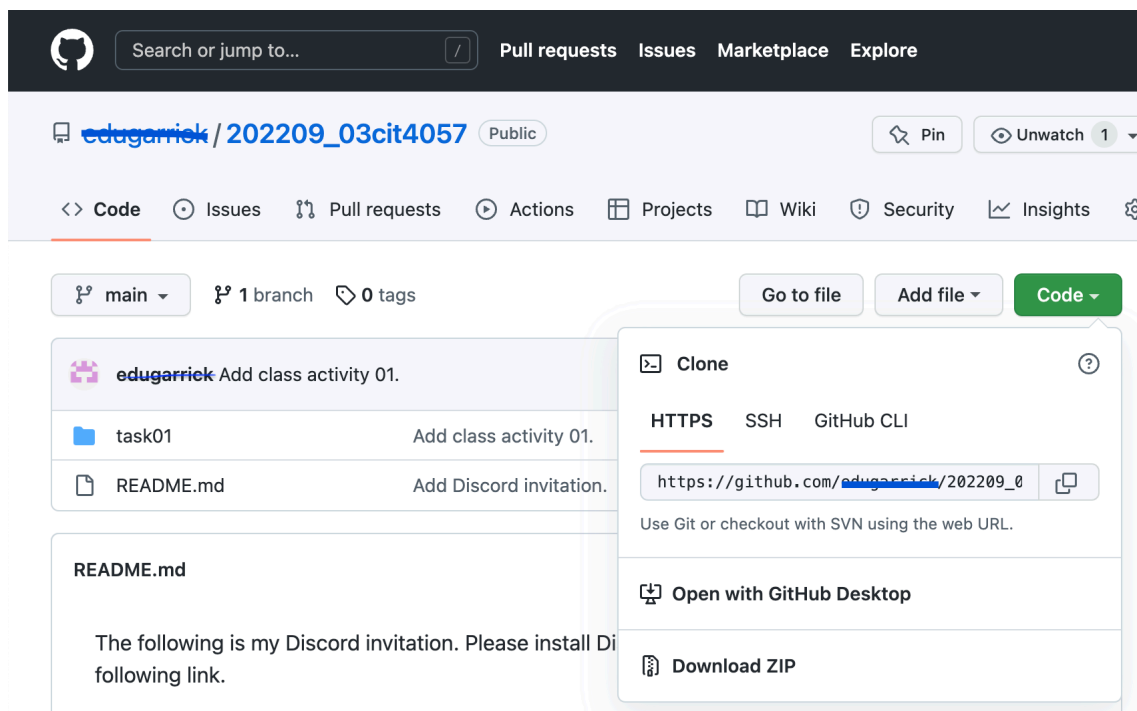
Classwork-1.1 : Clone the repository from GitHub

1. In GitHub, sync your forked repository.

2. In your PC, run the following command to clone the repository to your PC.

```
$ git clone <url>
```

where url is from



3. Make a directory under cw01/src. The directory name is your student id as below.

```
cw01
|_ src
|_ <your student>
```

Classwork-1.2 : Comment

1. In VSCode, type and save the following program as sumsquare.py.

```
# 1. Try comment
'''
This program calculates the sum of
square of first N natural
numbers
'''

def squaresum(n) :
    sum = 0
    for i in range(1, n+1) :
        sum = sum + (i * i)
    return sum

def main():
    integerN = 4
    # Try line wrap within parentheses
    print("The sum of square of integer N is ",
          squaresum(integerN))

if __name__ == "__main__":
    main()
```

3. You can run the program successfully.
4. Submit your work to your forked repository. The command is as followed.

```
$ git commit -m "Classwork-1.2." && git push
[main 36f90bd] Include Classword 01.
 1 file changed, 1 insertion(+)
...
Username for 'https://github.com': <your GitHub id>
Password for 'https://<your GitHub id>@github.com': <PAT>
```

Classwork-1.3 : Comparison Operator

1. In VSCode, type and save the following program as findmax.py.

```
# 1. Try indentation in a block of code
# 2. Try Comparison Operator

def maximum(num1, num2):
    if num1 >= num2:
        return num1
    else:
        return num2

def main():
    number1 = 1000
    number2 = 2919
    print(maximum(number1, number2))

if __name__ == "__main__":
    main()
```

2. You can run the program successfully.
3. Submit your work to your forked repository. The command is as followed.

```
$ git commit -m "Classwork-1.3." && git push
[main 36f90bd] Include Classword 01.
1 file changed, 1 insertion(+)
...
Username for 'https://github.com': <your GitHub id>
Password for 'https://<your GitHub id>@github.com': <PAT>
```

Classwork-1.4 : Arithmetic operator

1. In VSCode, type and save the following program as simpleinterest.py.

```
# 1. Try Arithmetic operator
def simple_interest(p,t,r):
    si = (p * t * r)/100
    return si

def main():
    principal = 8
    time = 6
    rate = 8
    print('The principal is', principal)
    print('The time period is', time)
    print('The rate of interest is', rate)
    # Try line wrap within parentheses
    si = simple_interest( principal, time,
                          rate)
    print('The Simple Interest is', si)

if __name__ == "__main__":
    main()
```

2. You can run the program successfully.
3. Submit your work to your forked repository. The command is as followed.

```
$ git commit -m "Classwork-1.4." && git push
[main 36f90bd] Include Classword 01.
1 file changed, 1 insertion(+)
...
Username for 'https://github.com': <your GitHub id>
Password for 'https://<your GitHub id>@github.com': <PAT>
```

Classwork-1.5: Submit your work by GitHub pull request

1. In the GitHub, create a pull request as shown in class activity 01.

Classwork-1.6 : Extra activity

1. Copy the handbook.txt to your student id directory.
2. In VSCode, run the following

```
pip3 install --upgrade pip
pip3 install pandas
pip3 install matplotlib
pip3 install wordcloud
```

3. In VSCode, type and save the following program as wc_gen.py.

```
# For fun and create Word Cloud.
# from a piece of text.
import pandas as pd
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS

df = pd.read_csv("handbook.txt", sep=" ")
text = " ".join( str(cat) for cat in df.Review)
word_cloud = WordCloud(
    width=3000,
    height=2000,
    random_state=1,
    background_color="salmon",
    colormap="Pastel1",
    collocations=False,
    stopwords=STOPWORDS,
).generate(text)

plt.imshow(word_cloud)
plt.axis("off")
plt.show()
```

3. You can run the program successfully and generate the word cloud as below.

