Course: CSC 220.02
Student: Kullathon “Mos” Sitthisarnwattanachai, **SFSU ID:** 921425216
Teammate: n/a, **SFSU ID:** n/a
Assignment Number: 01
Assignment Due Date & Time: 02-01-2021 at 11:55 PM

PART A – Communication, 10 points

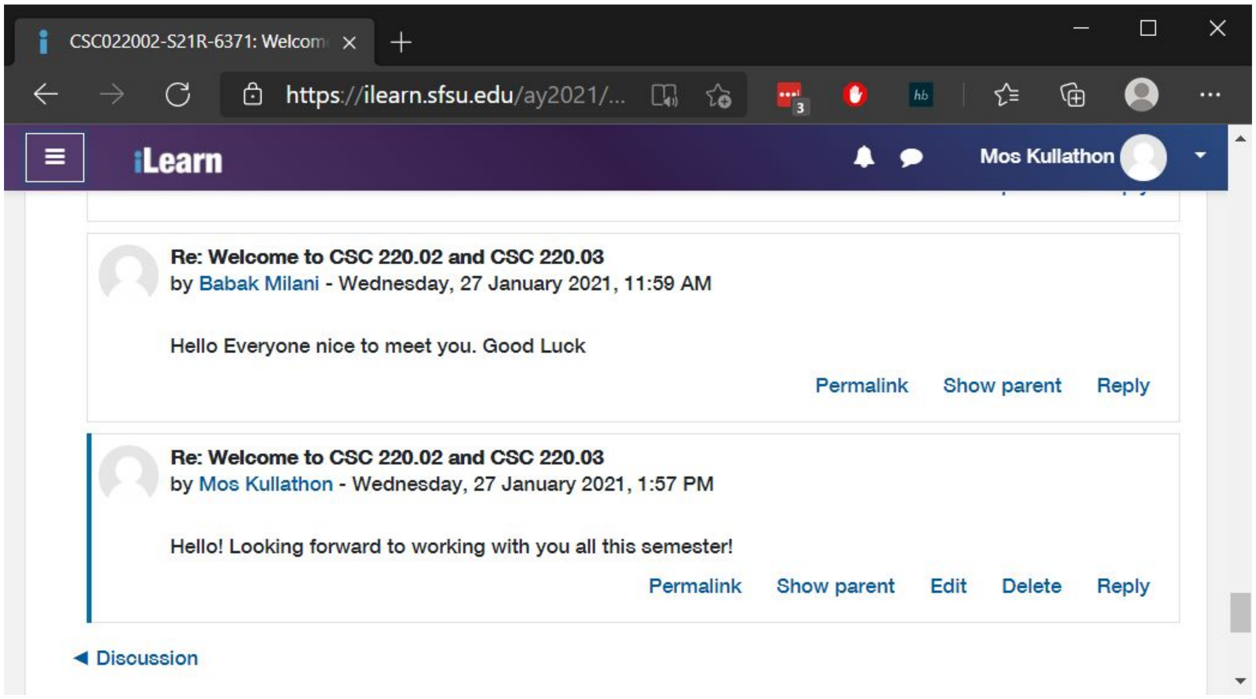
1. iLearn

    a. Log in iLearn.

    b. In “ASMT 1 Discussions” forum, please reply to the post “Welcome to CSC 220.02 and CSC 220.03”

    c. Take a screenshot of your reply.

    d. Include the screenshot in your assignment report.


CSC022002-S21R-6371: Welcom
https://ilearn.sfsu.edu/ay2021/...
iLearn
Mos Kullathon

Re: Welcome to CSC 220.02 and CSC 220.03
by Babak Milani - Wednesday, 27 January 2021, 11:59 AM

Hello Everyone nice to meet you. Good Luck

Permalink Show parent Reply

Re: Welcome to CSC 220.02 and CSC 220.03
by Mos Kullathon - Wednesday, 27 January 2021, 1:57 PM

Hello! Looking forward to working with you all this semester!

Permalink Show parent Edit Delete Reply

◀ Discussion

2. Emailing

    To answer A.2 questions, you may write: YES. This is to confirm that I …

a. Please write a statement here to confirm that you will use your SFSU email address (@mail.sfsu.edu) when contacting your grader and your course instructor so that your emails will not be filtered. Thank you.

   YES. This is to confirm that I will use my SFSU email address when contacting my grader and my course instructor.

b. Please write a statement here to confirm that when contacting your course instructor, you will start your email subject with this format: “CSC 220.02 PoNG” or “CSC 220.03 PiNG” so that you will get an answer timely. Thank you.

   YES. This is to confirm that I will use a subject line starting with “CSC 220.02 PoNG” for my email correspondence with the course instructor.

c. Please confirm that if you do not get an answer within 24 hours, you will check if you sent your email properly. And in either case, you will kindly resend your email message. Thank you.

   YES. This is to confirm that I will check if my email has been sent properly if I do not receive a response within 24 hours. And in either case, I will kindly resend the message.

3. Grader

We have one grader for each section of our class who grades our assignments. Questions regarding assignments and assignment grades should be directed to the grader.

a. What is the URL of your grader’s forum on iLearn?

   https://ilearn.sfsu.edu/ay2021/mod/forum/view.php?id=749561

b. What is your grader’s full name?

   PeiJun Huang

c. What is your grader’s SFSU email address? Please email your grader when you have questions.

   phuang5@mail.sfsu.edu

4. Guidelines for All Assignment and Assignment Report Template

   To answer A.4, you may write: YES. This is to confirm that I have carefully read, understood, and agreed to the Guidelines for ALL Assignments above and the Assignment Report Template. I will strictly follow the instructions.

   Please write a statement here to confirm that you have carefully read, understood, and agreed to the Guidelines for ALL Assignments above and the Assignment Report Template. If you have any questions, please list all of them here then please email the course grader or the course instructor before the 2nd week of the semester to get an answer. Thank you.

   YES. This is to confirm that I have carefully read, understood, and agreed to the Guidelines for ALL Assignments above and the Assignment Report Template. I will strictly follow the instructions.

5. Course Policy on Student Conduct and Academic Honesty

To answer A.5, please write exactly:

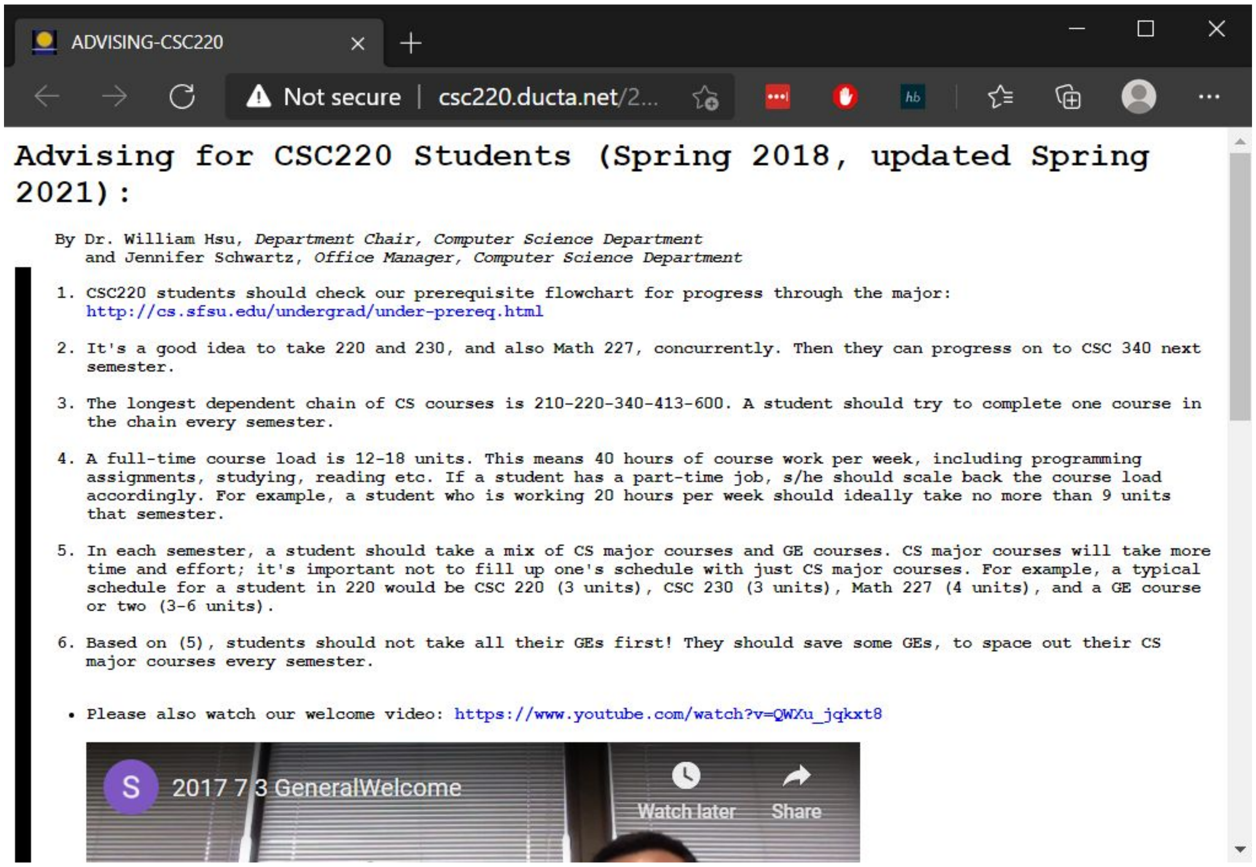
> YES. This is to confirm that I have carefully read, understood, and agreed to the Course Policy on Student Conduct and Academic Honesty which was distributed to me with the course syllabus and whose digital copy was shared with me on the File Manager. I am acutely aware that the policy includes, but is not limited to, the San Francisco State University's Code of Student Conduct (at https://conduct.sfsu.edu/standards), the Computer Science Department's Student Policies (at https://cs.sfsu.edu/student-policies), and the Honor Code of this course (at http://csc220.ducta.net/00-README-StudentConduct_AcademicHonesty.pdf). I will strictly follow all the rules.

Please write a statement here to confirm that you have carefully read, understood, and agreed to the Course Policy on Student Conduct and Academic Honesty which was distributed with the course syllabus and whose digital copy was shared on the File Manager. You are acutely aware that the policy includes, but is not limited to, the San Francisco State University's Code of Student Conduct (at https://conduct.sfsu.edu/standards), The Computer Science Department's Student Policies (at https://cs.sfsu.edu/student-policies), and the Honor Code of this course (at http://csc220.ducta.net/00-README-StudentConduct_AcademicHonesty.pdf). You will honor and strictly follow all the rules. If you have any questions, please list all of them here then please email the course grader or the course instructor before the 2nd week of the semester to get an answer. Thank you.

YES. This is to confirm that I have carefully read, understood, and agreed to the Course Policy on Student Conduct and Academic Honesty which was distributed to me with the course syllabus and whose digital copy was shared with me on the File Manager. I am acutely aware that the policy includes, but is not limited to, the San Francisco State University's Code of Student Conduct (at https://conduct.sfsu.edu/standards), the Computer Science Department's Student Policies (at https://cs.sfsu.edu/student-policies), and the Honor Code of this course (at http://csc220.ducta.net/00-README-StudentConduct_AcademicHonesty.pdf). I will strictly follow all the rules.

6. CSC220 File Manager
    a. Please log in to the CSC 220 File Manager.
    b. Locate file "2020Fall-CSC220-AdvisingForCSC220.html" and read through it. Include a screenshot of this page in your assignment report.

       *Perhaps the prompt meant "**2021Spring**-CSC220-AdvisingForCSC220.html"?*



Advising for CSC220 Students (Spring 2018, updated Spring 2021):

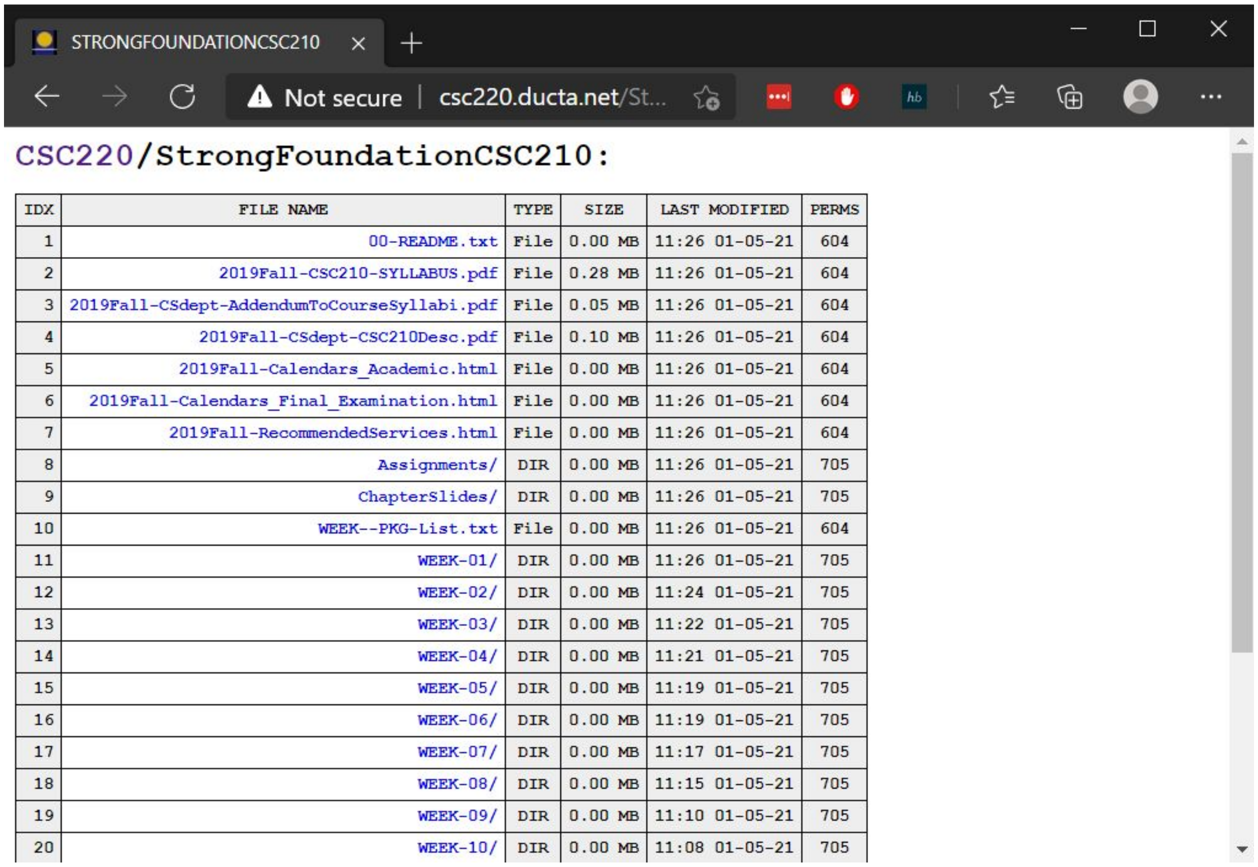
By Dr. William Hsu, *Department Chair, Computer Science Department*
and Jennifer Schwartz, *Office Manager, Computer Science Department*

1. CSC220 students should check our prerequisite flowchart for progress through the major: http://cs.sfsu.edu/undergrad/under-prereq.html
2. It's a good idea to take 220 and 230, and also Math 227, concurrently. Then they can progress on to CSC 340 next semester.
3. The longest dependent chain of CS courses is 210-220-340-413-600. A student should try to complete one course in the chain every semester.
4. A full-time course load is 12-18 units. This means 40 hours of course work per week, including programming assignments, studying, reading etc. If a student has a part-time job, s/he should scale back the course load accordingly. For example, a student who is working 20 hours per week should ideally take no more than 9 units that semester.
5. In each semester, a student should take a mix of CS major courses and GE courses. CS major courses will take more time and effort; it's important not to fill up one's schedule with just CS major courses. For example, a typical schedule for a student in 220 would be CSC 220 (3 units), CSC 230 (3 units), Math 227 (4 units), and a GE course or two (3-6 units).
6. Based on (5), students should not take all their GEs first! They should save some GEs, to space out their CS major courses every semester.

- Please also watch our welcome video: https://www.youtube.com/watch?v=QWXu_jqkxt8

c. Locate directory "CSC220/StrongFoundationCSC210" and browse through it.

   Include a screenshot of this directory in your assignment report.


STRONGFOUNDATIONCSC210

Not secure | csc220.ducta.net/St...

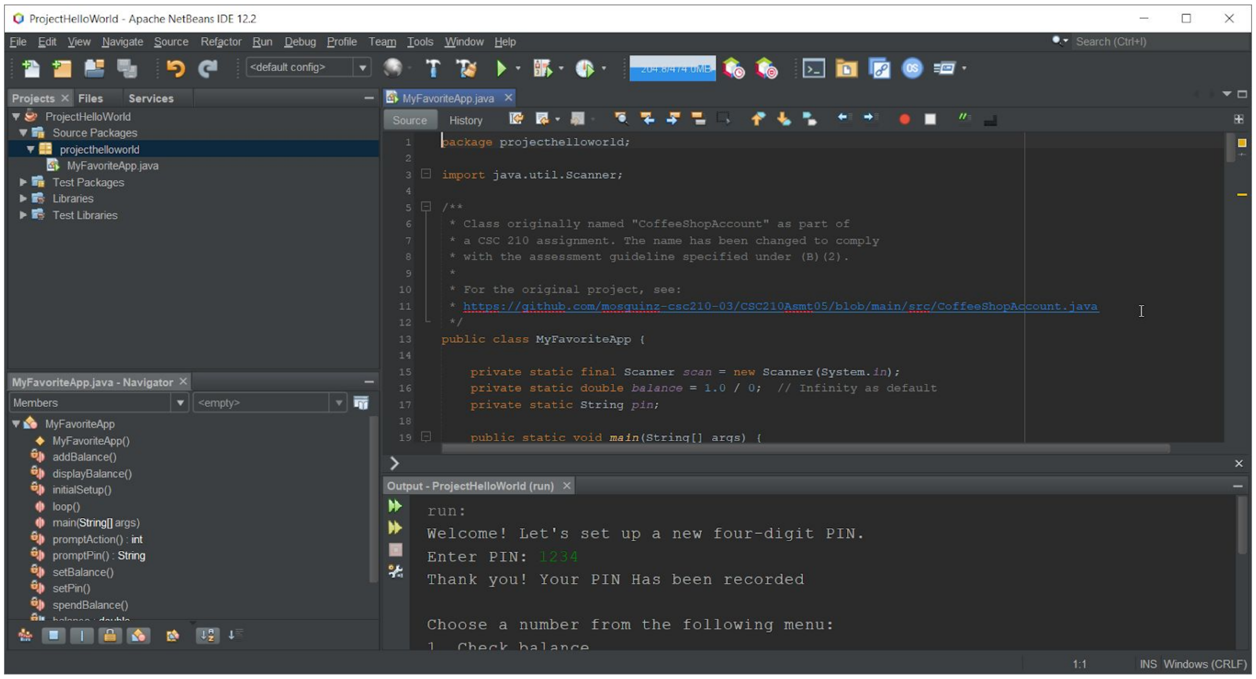
CSC220/StrongFoundationCSC210:

IDX	FILE NAME	TYPE	SIZE	LAST MODIFIED	PERMS
1	00-README.txt	File	0.00 MB	11:26 01-05-21	604
2	2019Fall-CSC210-SYLLABUS.pdf	File	0.28 MB	11:26 01-05-21	604
3	2019Fall-CSdept-AddendumToCourseSyllabi.pdf	File	0.05 MB	11:26 01-05-21	604
4	2019Fall-CSdept-CSC210Desc.pdf	File	0.10 MB	11:26 01-05-21	604
5	2019Fall-Calendars_Academic.html	File	0.00 MB	11:26 01-05-21	604
6	2019Fall-Calendars_Final_Examination.html	File	0.00 MB	11:26 01-05-21	604
7	2019Fall-RecommendedServices.html	File	0.00 MB	11:26 01-05-21	604
8	Assignments/	DIR	0.00 MB	11:26 01-05-21	705
9	ChapterSlides/	DIR	0.00 MB	11:26 01-05-21	705
10	WEEK--PKG-List.txt	File	0.00 MB	11:26 01-05-21	604
11	WEEK-01/	DIR	0.00 MB	11:26 01-05-21	705
12	WEEK-02/	DIR	0.00 MB	11:24 01-05-21	705
13	WEEK-03/	DIR	0.00 MB	11:22 01-05-21	705
14	WEEK-04/	DIR	0.00 MB	11:21 01-05-21	705
15	WEEK-05/	DIR	0.00 MB	11:19 01-05-21	705
16	WEEK-06/	DIR	0.00 MB	11:19 01-05-21	705
17	WEEK-07/	DIR	0.00 MB	11:17 01-05-21	705
18	WEEK-08/	DIR	0.00 MB	11:15 01-05-21	705
19	WEEK-09/	DIR	0.00 MB	11:10 01-05-21	705
20	WEEK-10/	DIR	0.00 MB	11:08 01-05-21	705

PART B – NetBeans IDE Installation, 5 points

1. Install the latest NetBeans IDE by following the instructions given in class and the lecture slides including:
   - On Windows:
     http://csc220.ducta.net/WEEK-01/NetBeans_Installation_Windows.pdf
   - On macOS:
     http://csc220.ducta.net/WEEK-01/NetBeans_Installation_MacOS.pdf
2. Change the name of a favorite Java program you wrote to “MyFavoriteApp.java” and run it in your newly installed NetBeans.

3. Take a screenshot of your NetBeans. The screenshot should show both code and output.



```java
package projecthelloworld;

import java.util.Scanner;

/**
 * Class originally named "CoffeeShopAccount" as part of
 * a CSC 210 assignment. The name has been changed to comply
 * with the assessment guideline specified under (B)(2).
 *
 * For the original project, see:
 * https://github.com/mosquinz-csc210-03/CSC210Asmt05/blob/main/src/CoffeeShopAccount.java
 */
public class MyFavoriteApp {

    private static final Scanner scan = new Scanner(System.in);
    private static double balance = 1.0 / 0;  // Infinity as default
    private static String pin;

    public static void main(String[] args) {
```

```
run:
Welcome! Let's set up a new four-digit PIN.
Enter PIN: 1234
Thank you! Your PIN Has been recorded

Choose a number from the following menu:
1. Check balance
```

In half a page or more, outline how you would make your program better.

The program was made as part of an assessment during my CSC 210 class last semester. So naturally, I had to strictly follow the assessment guideline and was working under a relatively tight time constraint.

The program was very basic and did not contain a lot of features. This program was made to mimic an account for a coffee shop. One of the biggest issues I would address is input sanitation. The program reads in the user input from stdin without really validating it (with the exception of the PIN). Since much of the program relies on the Scanner to read and automatically cast the input into certain data types, the Scanner will throw an exception if it receives an input of an incorrect type. For example, If the end user enters a string “abc” when prompted for an amount of money. The program did not consider any of this since we were instructed to ignore invalid input. Any input that would cause the Scanner to throw an exception would cause the program to crash. If I had time to improve it, I would consider handling the exception thrown by the Scanner and print out appropriate messages, so that the user can alter their input accordingly.

Another way I would improve the program would be to write the user data into files. At its current state, none of the information is saved when the program is run. The program would lose all of its information each time it restarts as it has no way to store it. Since the assessment did not require us to save the information in any way, all of the information is only stored in variables in each instance of the application. I could improve the program by creating a way to write the value to a file, and a way for the program to read those values later on.

4. Please remember to submit your MyFavoriteApp.java file (and related files, if any).

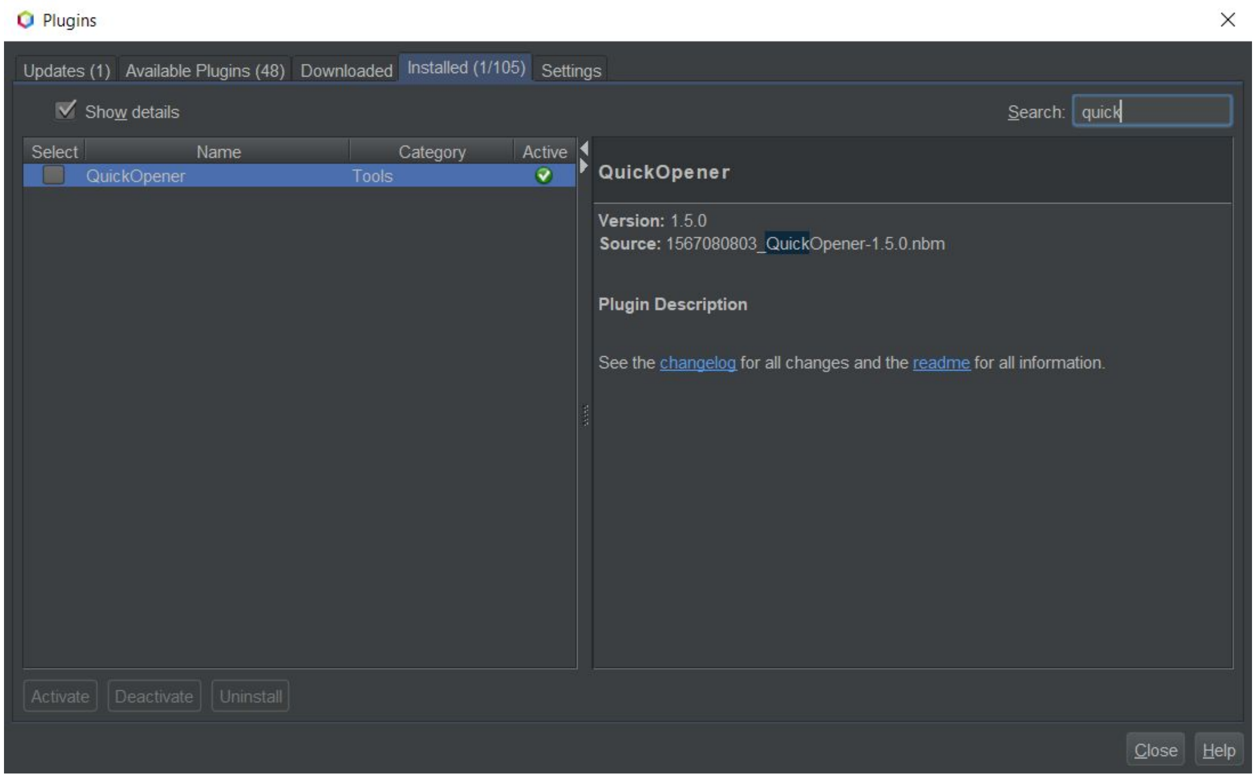
PART C – NetBeans Plugins and Shortcuts, 5 points

1. Install 3 NetBeans Plugins
    - Please install QuickOpener and 2 other plugins of your choice.
    - Useful links, applicable to the newer NetBeans versions including NetBeans 12.2 of 2021:
        - http://csc220.ducta.net/WEEK-01/NetBeans_Installation_QuickOpener_Plugin.pdf

- https://cwiki.apache.org/confluence/display/NETBEANS/Using+8.2+Plugins+in+9.0
- http://plugins.netbeans.org/plugin/62668/quickopener

a. In your NetBeans, take screenshot(s) of the window Tools/Plugins/Installed to show the 3 plugins you installed.

The screenshots below shows the QuickOpener, gitignore.io, and Darcula LAF for NetBeans plugin in the "Installed" tab of my NetBeans installation.



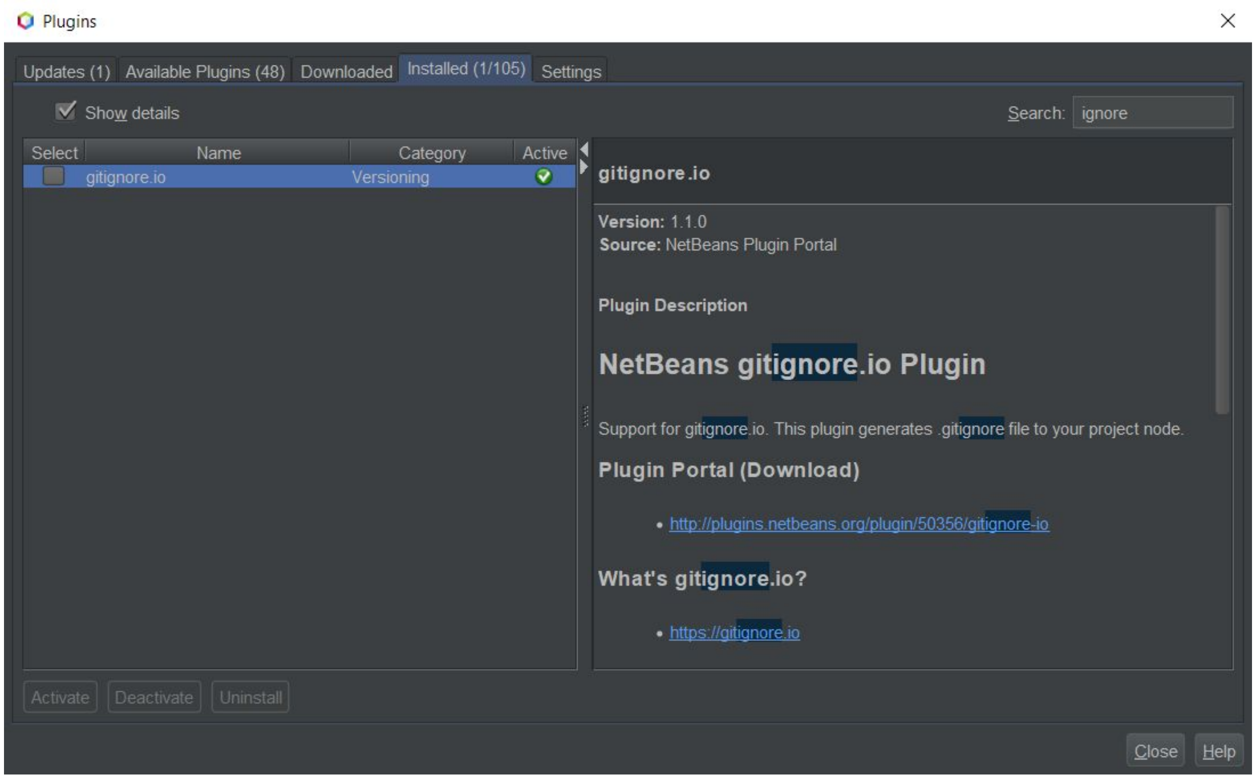
Select	Name	Category	Active
☐	QuickOpener	Tools	✓

QuickOpener

Version: 1.5.0
Source: 1567080803_QuickOpener-1.5.0.nbm

Plugin Description

See the changelog for all changes and the readme for all information.


Plugins

Updates (1) | Available Plugins (48) | Downloaded | Installed (1/105) | Settings

Show details — Search: ignore

Select	Name	Category	Active
☐	gitignore.io	Versioning	✓

gitignore.io

Version: 1.1.0
Source: NetBeans Plugin Portal

Plugin Description

NetBeans gitignore.io Plugin

Support for gitignore.io. This plugin generates .gitignore file to your project node.

Plugin Portal (Download)

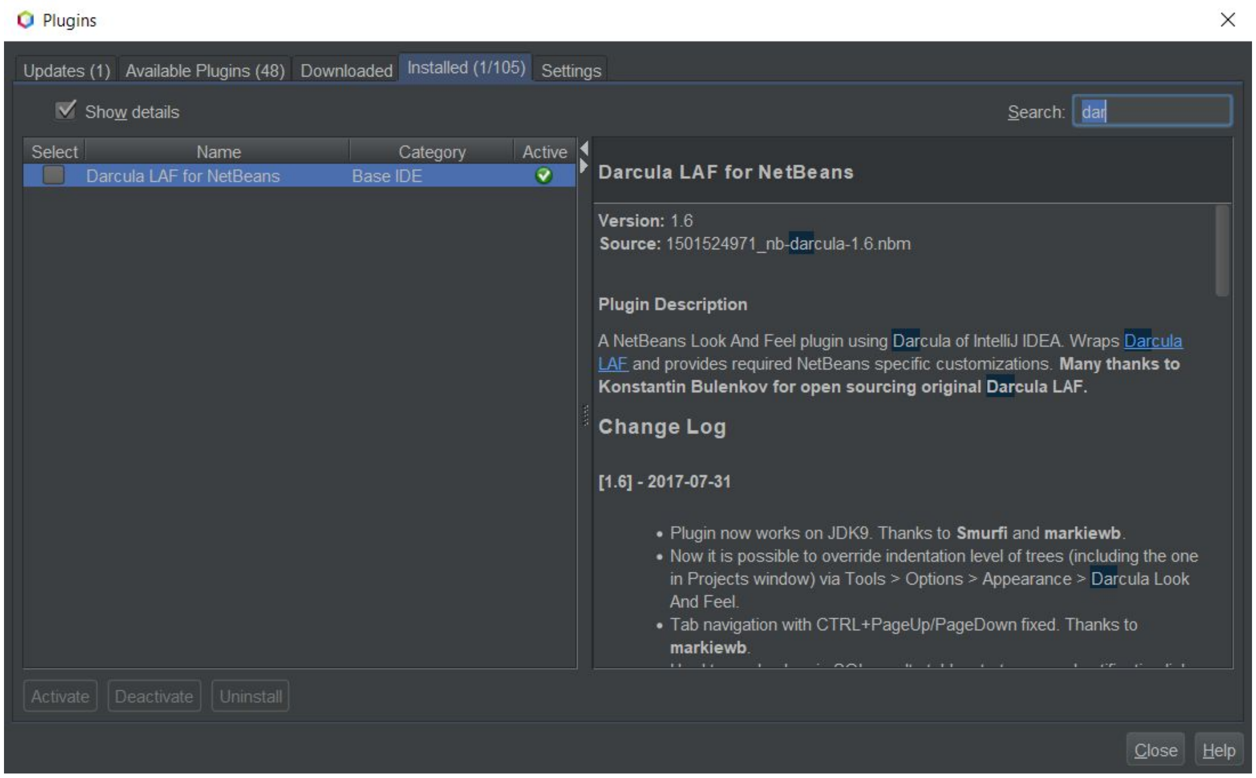
- http://plugins.netbeans.org/plugin/50356/gitignore-io

What's gitignore.io?

- https://gitignore.io

Activate | Deactivate | Uninstall

Close | Help


Plugins

Updates (1) | Available Plugins (48) | Downloaded | Installed (1/105) | Settings

Show details — Search: dar

Select	Name	Category	Active
☐	Darcula LAF for NetBeans	Base IDE	✓

Darcula LAF for NetBeans

Version: 1.6
Source: 1501524971_nb-darcula-1.6.nbm

Plugin Description

A NetBeans Look And Feel plugin using Darcula of IntelliJ IDEA. Wraps Darcula LAF and provides required NetBeans specific customizations. **Many thanks to Konstantin Bulenkov for open sourcing original Darcula LAF.**

Change Log

[1.6] - 2017-07-31

- Plugin now works on JDK9. Thanks to **Smurfi** and **markiewb**.
- Now it is possible to override indentation level of trees (including the one in Projects window) via Tools > Options > Appearance > Darcula Look And Feel.
- Tab navigation with CTRL+PageUp/PageDown fixed. Thanks to **markiewb**.

Activate | Deactivate | Uninstall

Close | Help

b. In 4 sentences or more, discuss the 2 plugins you chose.

The gitignore.io plugin comes from the gitignore.io website that I (and perhaps you, too) frequently use. It generates a .gitignore file for the specific language and environment(s) that you use. The plugin simply extends that feature and puts the file directly into your project's directory.

The Darcula LAF for NetBeans plugin changes the appearance of NetBeans. It changes the color theme and syntax highlighting that closely resembles those of IntelliJ's dark theme, so I can pretend that I am using IntelliJ (kind of).

2. Netbeans-Shortcuts-80.pdf

   a. In the CSC 220 File Manager and under WEEK-01 directory, open file "Netbeans-Shortcuts-80.pdf"

   b. Try as many NetBeans shortcuts as you can and see if they work in the latest NetBeans you installed.

   c. In 3 sentences or more, describe your favorite shortcut(s).

Rectangular selection (Ctrl + Shift + R) is very helpful for selecting patterned code. For example, if you need to edit something at the end of the line for multiple lines. Similarly, the add caret mode (Ctrl + Shift + Click) also allows you to type in multiple places at once. Another closely related shortcut I can't live without is the select next occurrence (Ctri + J), which does exactly that. Very handy for quick rename-and-refactor in small places. It also adds a caret for each selection it makes, so you are able to edit in multiple places.

PART D – Class Design Guidelines, 2 Extra Credit points

The WEEK-01 lectures introduce Y. Daniel Liang's Class Design Guidelines, in 1 page or more, discuss in-depth 1 of the guidelines. You can use code to demonstrate your points. The code should not be more than one-third of your writing.

Consistency is one of the principles in Liang's class design guidelines. It dictates the use of naming conventions and programming styles. Consistent naming convention allows the code to be understood and utilized by others. This is important when a project is developed by a group of people. Having a set of conventions agreed upon and adhered to allows the programmers to quickly identify the purpose of the code and keep the project maintainable. For example, the names of variables and methods in Java are lowerCamelCased, whereas the class names are written in UpperCamelCase. Aside from the casing, the actual names of these components are also important. For example, an integer variable declared as `int x;` does not provide any useful information to the programmers you work with (and after some time, yourself) and any context as to what x actually does. Instead, a more descriptive name could be used in its declaration, such as `int numberOfClients;`. Simple rules like this can allow you and other programmers to keep track of your variables and can save a lot of time when trying to identify an issue.

Other styling practices such as tab sizes, indentation, and ordering should also be followed for the sake of consistency between classes. For example, in Java field declarations of a class are often at the top (with constants being at the top of those), followed by its constructors, then its methods (where the applicable setters and getters are placed first). While these conventions are not enforced by the compiler -- meaning the code will still function if you put your instance methods at the top of the class and name your constants in lowercase -- it is still important to follow such conventions as it allows different parts of the class and their purposes to be quickly identified. Having different classes structured in a similar manner allows for a much more effortless design process and can also prevent common errors from arising.