EGR 222 — Software Engineering
Homework #1 Rocket Ship
Check Point Due: Friday, 9/4/20, 11 PM
Final Due: Tuesday, 9/8/20, 11 PM

[Section 1: Specification]

This assignment will give you practice with for loops, static methods, print/println statements and a class constant.

In this assignment, you are going to generate output that looks like the following:

```
     /*++*\
    //*++*\\
   ///*++*\\\
  ////*++*\\\\
 /////*++*\\\\\
**=+=+=+=+=+=+**
||\/\/\/\/\/\/||
||.\/\/..\/\/.||
||..\/....\/..||
||../\..../\..||
||./\/\../\/\.||
||/\/\/\/\/\/\||
**=+=+=+=+=+=+**
||../\..../\..||
||./\/\../\/\.||
||/\/\/\/\/\/\||
||\/\/\/\/\/\/||
||.\/\/..\/\/.||
||..\/....\/..||
**=+=+=+=+=+=+**
     /*++*\
    //*++*\\
   ///*++*\\\
  ////*++*\\\\
 /////*++*\\\\\
```

You are to exactly reproduce the output given under ***expected_output*** directory of the starter code given. The various subfigures in the middle of above output have a height of 3. These subfigures have the property that their height determines their width, so there is only one size variable. You are to use a class constant to make it possible to change a single number in one place in the program to have it produce a corresponding figure of a different size. You can **more example output with varying size under *expected_output* directory of the starter code given (See Section 2).**

This assignment is meant to give you practice with the constructs from chapters 1 and 2. This will require you to create nested for loops with print and println statements that use the class constant. You may use constructs from chapter 3, although you are not required to do so and you will receive no extra credit for doing so. You may not use any programming constructs that are not in chapters 1 through 3 of the textbook.

You should continue to use static methods to structure your solution. You should try to avoid significant redundancy and you should structure your program in such a way that the methods match the structure of the output itself. You are required to properly indent your code and will lose points if you make indentation mistakes. You should localize variables whenever possible.

You should once again include a comment at the beginning of your class file with basic information and a description of what the program does. You should also include a comment on each individual method describing what it does. You will implement on **DrawRocket.java**. On any given execution your program will produce just one version of this figure, but it should be possible to change the value of the program constant to have your program produce a figure of a different size. For example, with subfigures of height 5, the output should look like below. (Since it is hard to see the number of spaces from this document, do not use this figure but make sure you **refer to the files under *expected_output* directory of the starter code**).

```
         /*++*\
        //*++*\\
       ///*++*\\\
      ////*++*\\\\
     /////*++*\\\\\
    //////*++*\\\\\\
   ///////*++*\\\\\\\
  ////////*++*\\\\\\\\
 /////////*++*\\\\\\\\\
**=+=+=+=+=+=+=+=+=+=+**
||\/\/\/\/\/\/\/\/\/\/||
||.\/\/\/\/..\/\/\/\/.||
||..\/\/\/....\/\/\/..||
||...\/\/......\/\/...||
||....\/........\/....||
||..../\......../\....||
||.../\/\....../\/\...||
||../\/\/\..../\/\/\..||
||./\/\/\/\../\/\/\/\.||
||/\/\/\/\/\/\/\/\/\/\||
**=+=+=+=+=+=+=+=+=+=+**
||..../\......../\....||
||.../\/\....../\/\...||
||../\/\/\..../\/\/\..||
||./\/\/\/\../\/\/\/\.||
||/\/\/\/\/\/\/\/\/\/\||
||\/\/\/\/\/\/\/\/\/\/||
||.\/\/\/\/..\/\/\/\/.||
||..\/\/\/....\/\/\/..||
||...\/\/......\/\/...||
||....\/........\/....||
**=+=+=+=+=+=+=+=+=+=+**
         /*++*\
        //*++*\\
       ///*++*\\\
      ////*++*\\\\
     /////*++*\\\\\
    //////*++*\\\\\\
   ///////*++*\\\\\\\
  ////////*++*\\\\\\\\
 /////////*++*\\\\\\\\\
```

[Section 2: Project Setup Instruction]

0. Check your SDK version in IntelliJ by going to File → Project Structure (Or short-cut, Ctrl+Alt+Shift+S). The version info is under **Project SDK** and **Project language level**. Make sure you have the latest version.

1. Click below and accept the invitation if you haven't.

https://classroom.github.com/a/V-MeM4cy

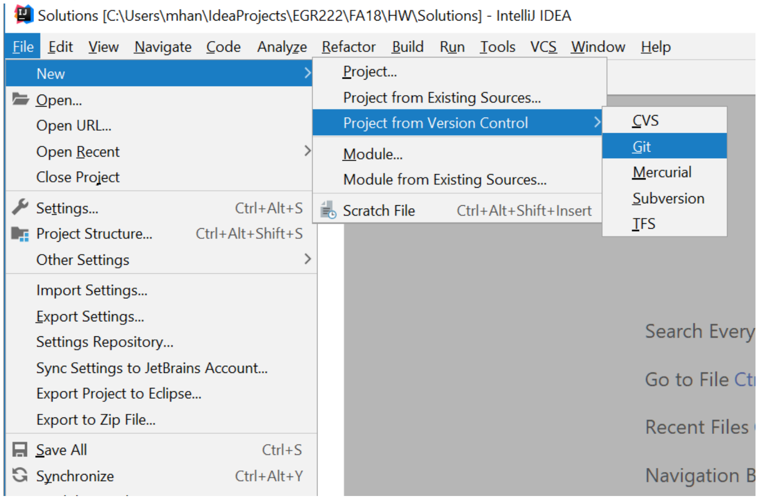
2. Go to your repository where the URL is

https://github.com/cbu-egr222-fa20/hw1-{YourGithubID}

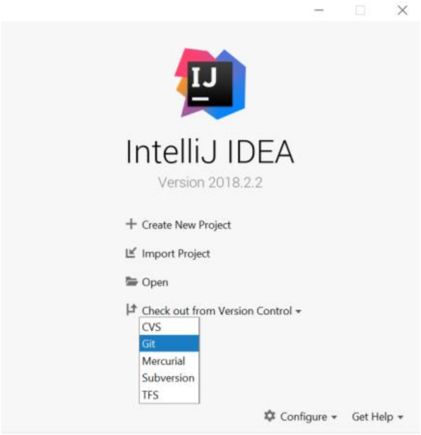
Check you have .idea/ and src/ and other support files mentioned in Section 1.

3. Open IntelliJ (Do A or B depending on what you see there)

A) If you see below UI, go to File → New → Project from Version Control → Git


Solutions [C:\Users\mhan\IdeaProjects\EGR222\FA18\HW\Solutions] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
New
Open...
Open URL...
Open Recent
Close Project
Settings... Ctrl+Alt+S
Project Structure... Ctrl+Alt+Shift+S
Other Settings
Import Settings...
Export Settings...
Settings Repository...
Sync Settings to JetBrains Account...
Export Project to Eclipse...
Export to Zip File...
Save All Ctrl+S
Synchronize Ctrl+Alt+Y
Project...
Project from Existing Sources...
Project from Version Control
Module...
Module from Existing Sources...
Scratch File Ctrl+Alt+Shift+Insert
CVS
Git
Mercurial
Subversion
TFS
Search Every
Go to File Ct
Recent Files
Navigation B

B) If you see below window, select "Check out from Version Control → Git"

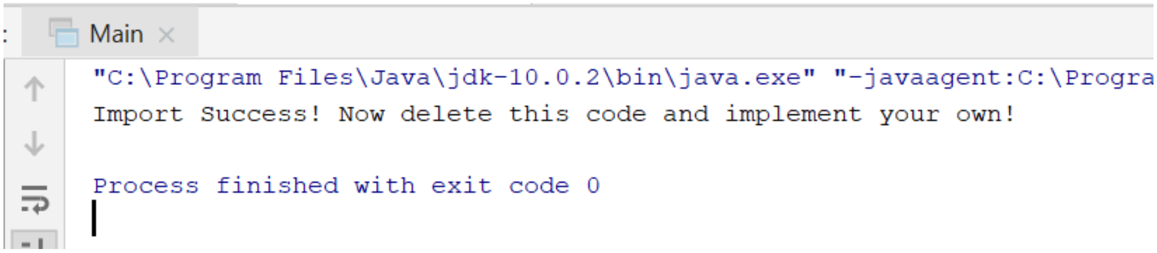


3. Put the URL for your repository and hit Clone. Again, the URL is

https://github.com/cbu-egr222-fa20/hw1-{YourGithubID}

4. In the “Project” pane make sure you see all the contents the same as your Github remote repository in 2.

5. Run main of this project and make sure you see below output.



```
"C:\Program Files\Java\jdk-10.0.2\bin\java.exe" "-javaagent:C:\Progra
Import Success! Now delete this code and implement your own!

Process finished with exit code 0
```

6. You are ready to implement your code!

[Section 3: Submission Instruction]

By Check Point Deadline, push your code progress to Github.

By Final Deadline, push your final code to Github and submit a Word doc with screenshots to BB.

When you are done implementing code, submit to Github. The latest committed version before the deadline will be used for grading. You must match the expected output exactly for each size given.

1. Go to your terminal in IntelliJ and type below commands.

```
> git add --all
> git commit -m “Put your comments here”
> git push
```

If you need to make some modification and resubmit, you can simply repeat above 3 commands again.

2. **THIS IS THE MOST IMPORTANT STEP**

Go to your remote repository where the URL is

https://github.com/cbu-egr222-fa20/hw1-{YourGithubID}

Navigate to **src** directory from your web browser and TAKE A SCREENSHOT of your source code that shows your main method. TAKE EXTRA TIME to make sure you can view your latest changes here. Again, the latest committed version you see here before the deadline will be used for grading. After you double check the push was successful, **include the screenshot of your source code that has main in a Word doc and submit to BB.**