Contents

[1.1 Question 3](#_Toc517999516)

[1.2 Instructions to Follow 3](#_Toc517999518)

[1.3 GIT Structure 3](#_Toc517999519)

[1.3 Instructions about the README.md file 4](#_Toc517999522)

[1.3.1 About README.md 4](#_Toc517999523)

[1.3.2 README.md template 4](#_Toc517999524)

[1.3.3 README.md Images 6](#_Toc517999525)

# Question

On a conveyor belt in a factory there are two types of unfinished goods of certain quantity rotating in alternating sequence of:

* Bolts
* Machine

There are 2 bolts which get fixed into one machine and produce a final product. The firm has 3 employees who pick up one raw material at a time (i.e. the person can either pickup one bolt or one machine). Once when the employee has the right quantity of raw material to create a product,  he goes on to create the product in a room for which he takes x minutes and then comes back again to start from picking the material from the belt.

The supervisor now wants to clear off all the pending raw material in the stock. He wants to create a program which will help understand within much time he would be able to consume all the raw material.

Taking the inputs as:

1. X (integer) which defines the total machines
2. Y (Integer) which defines the total bolts
3. N(Time in seconds) to assemble a product

The expected output should be the total time (In seconds) needed to utilize all the available raw material and create the products and the total number of products.

So for e.g.

Given time to assemble a product is 1 minute then, if there are 3 machines and 6 bolts in the warehouse then the employees would take 1 minute to create 3 products.

Hint Note: If the other worker thread accidentally picks wrong quantity of bolts   then there could even be 0.

In terms of Java program think of these three people as three different worker thread.

1. Create a function via which you can pick up the unfinished good one at a time.
2. Create a function which combines both the two unfinished good to create one product. Has a sleep for N seconds

**Input:**

X = 3

Y = 6

N = 60

**Sample Output:**

Total Products = 3

Total Time Taken = 60

# Instructions to Follow

1. Your code should be executable without any errors
2. Your code should be having proper test coverages
3. Your code should be using either Gradle or Maven as the build tool.
4. You could use any IDE of your choice, however the code has to be compatible with JDK 1.7 or greater
5. You would be requiring one of the code quality tools like Sonar Lint/EclEmma
6. You can check-in solutions as many times as you'd like, there are no penalties for incorrect submissions.
7. As a candidate, you are responsible for making sure others don't access the code that you submit.
8. Your code should not be using any software which has licensing or carries any copyrights
9. Provide access to the javasrassoc in GitHub if the code is not public.

# GIT Structure

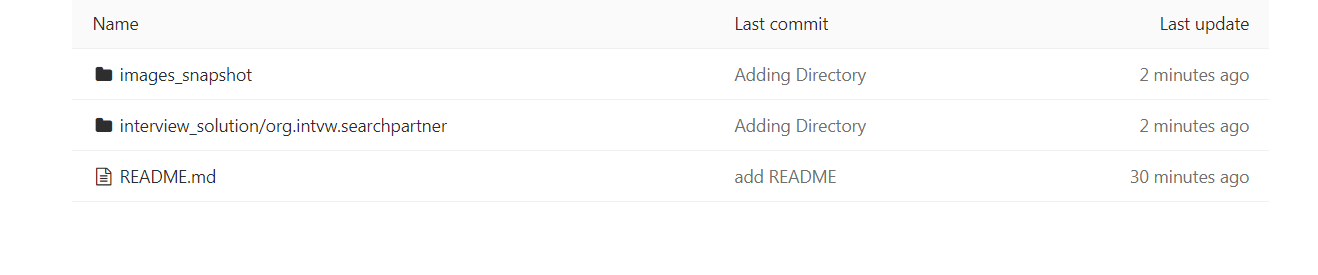
1. Link for the GIT where you are uploading the code should be like:

[**https://github.com/yourlogin/intrvw\_searchpartner**](https://github.com/yourlogin/intrvw_searchpartner) **or https://github.com/yourlogin/somethingunique**

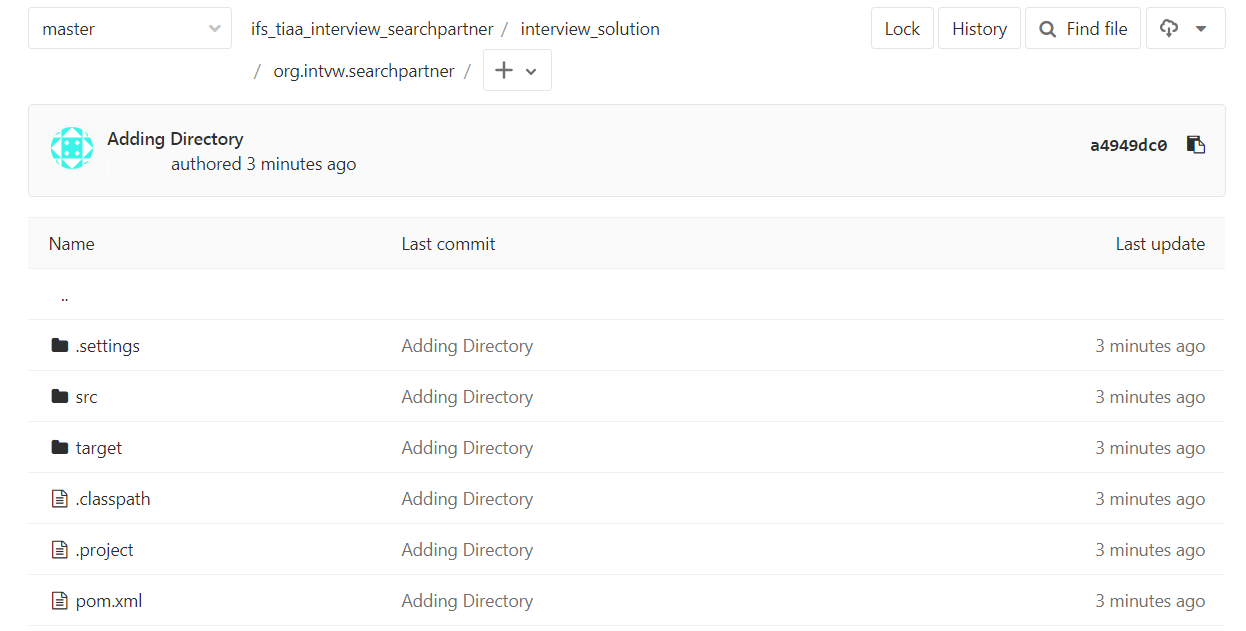
1. So in case your login id is nescafe2018 and you are via the search partner search quest the link should

[**https://github.com/nescafe2018/intrvw\_sq**](https://github.com/nescafe2018/intrvw_sq)

Your GIT structure would be something like below



1. One of the most crucial things in your solution would be the project structure. Your project structure should look something like this when you have checked in the code in GIT repository.



# Instructions about the README.md file

## About README.md

1. At the top of the file which is the first section there should be a short introduction about you mentioning your name and years of experience.
2. Second Section should be focusing on the BUILD IDE which you have used to code and the JDK version.
3. Third Section should be about SonarLint/EclEmma and the output
4. Fourth Section should be about the images names in the image folder which helps the panel to verify the output/details from the third section
5. Fifth Section should be the Licensing.

## README.md template

Section 1

Name: *First Name Last Name*

Years of Experience: *X years Y months*

Section 2:

Build IDE: *Eclipse Java EE IDE (Version: Oxygen.3a Release (4.7.3a))*

JDK Version: *java version "1.8.0\_162"*

Section 3

1. Sonar lint / EclEmma scan Snapshot
2. Final Output Snapshot
3. Junit Coverage Snapshot

*Note: (You can link directly to the raw version of an image from your image folder. Hence from section 1.1 if your GIT Link is :* [*https://github.com/nescafe2018/intrvw\_sq*](https://github.com/nescafe2018/intrvw_sq) *, then you can have the following text in your readme ![SonarLint] (https://github.com/nescafe2018/intrvw\_sq/<Link of you image) )*

Section 4

*Licensing*

1. The code does not have any Copyright whatsoever.
2. The code IS NOT governed by any licenses whatsoever.
3. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR USERS OF THE CODE WOULD BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## README.md Images

## 

**Fig 1.2.2.1**

## 

**Fig 1.2.2.2**