

Git Assignment

DEADLINE: Thursday, September 3, 2015, 11:59PM.

1. Create a GitHub account if you don't have one already.
2. For this assignment you'll need to perform several operations to familiarize yourself with Git and GitHub, as well as write a small application in either C++ or Java, using the contents of one of the repositories below (choose one):
For Java, access the following GitHub repository : <https://github.com/optimus9p/Oracle>
For C++, access the following GitHub repository : <https://github.com/optimus9p/OracleC>
3. Fork the selected project using the GitHub fork option. This will result in you obtaining your own copy of the repository and all its contents.

Forking optimus9p/Oracle

It should only take a few seconds.



4. Download the forked repository into your local machine using either a client or the console. The following steps are done in your local copy of the repository.
5. Remove the file REMOVE.txt from the repository. (5 points)
6. Modify the file README.txt to include your name and computer science username in the first two lines. (5 points)

Example:

Name: Mark Rivers

CS username: mrivers

7. Commit these changes to the repository using the following template (5 points):
Subject: A short description of the commit
Content describing in a concise way what was done, how and why.
Follow the guidelines presented at: <http://chris.beams.io/posts/git-commit/>
Example:
Update information files

The information files have been updated to describe the current state of the repository.
Repository owner information added to README.txt and REMOVE.txt deleted.
8. Create a new file named “book.txt”. Add to this file a new paragraph. Add the file to the repository. (5 points)
9. Use the command `git status`. Append the output of this command as content of the book.txt file. Commit the changes to the repository. (5 points)
10. Synchronize your local repository with your GitHub repository.
11. Create a new file named “answers.txt”. Add it to the repository. (5 points)
12. Answer the following questions in the answers.txt file, and then commit the changed file to the repository:
 - (a) How many commits have been done to the repository? (5 points)
 - (b) When was the file .gitignore modified last? (5 points)
 - (c) List the files, which were in the repository on 08/21/2015 at 11:00PM. (5 points)
 - (d) How many times was each of the files inside the src folder modified? (5 points)
 - (e) What is the difference between `git log` and `git status`? (5 points)
 - (f) What command would you use to see the commits where “main.java” was one of the committed files? (5 points)
 - (g) What command would you use to see the commits whose commit message contains the word “transaction”? (5 points)
 - (h) What command would you use to see the commits made by “Lauren” between 01/05/2014 and 06/14/2015? (5 points)
13. Use the content of the forked repository as a basis to implement the Hardwood Seller program described below. Synchronize and commit your changes to the repository frequently, after each significant change and write meaningful commit messages every time. Make sure you commit all your valid changes, and double check that everything was properly committed. If it’s not in the repo, it does not count. (30 points)

Hardwood Seller

The company VeLuvana Inc. is a hardwood wholesaler that has hired you to develop a program to compute the delivery time for the received orders. Currently the company determines the delivery time of their new orders manually. But due to the increase in the number of orders being received this is no longer viable.

The company handles five kinds of hardwood and sawdust; wood types and their respective prices per Board Foot (BF) are presented in Table 1:

Hardwood Type Price	Price
Cherry	\$5.95
Curly Maple	\$6.00
Genuine Mahogany	\$9.60
Wenge	\$22.35
White Oak	\$6.70
Sawdust	\$1.5

Table 1: Prices table

Note: Prices are per Board Foot (BF), the company uses only dollars for its transactions.

The company determines the delivery time based on the type and amount of wood. Depending on the type of wood, a base delivery time is set, which is later combined with the amount of wood purchased to compute the delivery time for each item in the order. The company places the order with the highest overall delivery time.

The base delivery times are presented in Table 2.

Hardwood Type	Delivery Time (hours)
Cherry	2.5
Curly Maple	1.5
Genuine Mahogany	3
Wenge	5
White Oak	2.3
Sawdust	1

Table 2: Base delivery times

The shipping time is calculated per each 100 BF, in hours as follows:

$$\begin{aligned}
[1, 100] &= 1 * \text{base delivery time} \\
[101, 200] &= 2 * \text{base delivery time} \\
[201, 300] &= 3 * \text{base delivery time} \\
[301, 400] &= 4 * \text{base delivery time} \\
[401, 500] &= 5 * \text{base delivery time} \\
[501, 1000] &= 5.5 * \text{base delivery time}
\end{aligned}$$

Example: A customer places an order with the following items: 100 BF of Cherry, 250 BF of Wenge, 755 BF of sawdust and 120 BF of White Oak. The delivery time for the order will be computed as follows:

$$\begin{aligned}
Cherry &= 1 * 2.5 = 2.5 \text{ hours} \\
Wenge &= 3 * 5 = 15 \text{ hours} \\
Sawdust &= 5.5 * 1 = 5.5 \text{ hours} \\
WhiteOak &= 2 * 2.3 = 4.6 \text{ hours} \\
DeliveryTime &= \max \{Cherry, Wenge, Sawdust, WhiteOak\} \\
&= \max \{2.5, 15, 5.5, 4.6\} \\
DeliveryTime &= 15 \text{ Hours}
\end{aligned}$$

In the requirements meeting the customer specified the following requirements for the software:

- The software would read the order from a text file with the following format:
 <Buyer full name>;<Address>;<Date>
 <WoodType>:<Amount>;<WoodType>:<Amount>; ... ;<WoodType>:<Amount>

Example:

Florida State University;600 W. College Avenue, Tallahassee, FL 32306;08/20/2015
 White Oak:500;Wenge:520

- The software should handle from 0 up to 1000 BF per each kind of wood.
- The output should contain the following data: full name of the buyer, address of delivery, list of the ordered wood, including BF purchased and its price, the estimated delivery time and total price of the purchase.

DELIVERABLES:

The contents of the GitHub repository by the deadline (remember to synchronize and your repository!!). The last committed changes in the repository by the deadline must be the final version of the software, any changes after the deadline will not be considered.