

Unit 1: Primitive Types

Topic 5 Lab 2: Partner Programming Challenges!

Name: _____


This is a **pair programming** lab: you and your partner should code together in the **same** Replit project, just like if you were both typing in a shared Google Doc!


Each partner should submit their own version of this Google Doc, but you and your partner should have copied/pasted the same code!

Setup

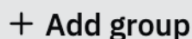
Follow the instructions below to create a collaborative Replit for you and your partner.

Choose **ONE** partner to:

A. Open up the  **U1T5 Lab 2: Partner Challenges** a shared Replit for you and your partner to co... team project.

B. Click:  on the right

C. Click **Add Group**:



D. Click **OK**:

Do you want to create a new project group?


You will not be able to leave or change project groups without assistance from the team admin.


 Cancel

 Ok

E. Click **Confirm** to allow notifications (if this pop up appears).

After the first partner completes the steps above, the **OTHER** Partner should:

A. Open up the  **U1T5 Lab 2: Partner Challenges** a shared Replit for you and your partner to co... team project.

B. Click:  on the right

- C. Locate and **Join** the group *started by your partner!* Make sure you select the right one (if you accidentally join the wrong group, let Mr. Miller know):

Select project group

@Lena

Once both partners are in, both partners will be coding alongside each other!

NOTE! For some of these challenges, you might need to use an `if-else` **selection statement**. We will discuss this statement and Boolean logic in depth in the future, but for now, here is the syntax and an example:

if-else statement example

This is a "2 way branch" -- *either one branch or the other will execute (but never both)*

```
int x = 3;
System.out.println("Hello!");

if (x > 4) {
    System.out.println(x + " is greater than 4!");
} else {
    System.out.println(x + " is not greater than 4!");
}

System.out.println("Goodbye!");
```

Programming Challenge 1: Test Average Calculator

Write a program to accept 4 integer test scores from the user (e.g. 75, 80, 82, and 90) and also accept an integer “passing score” value (e.g. 65).



Have your program print the four test scores (as integers), as well as calculate and print the average of the test scores (as a double). Then print a message to the user that informs them if the average test score is passing or not.

Don't forget about casting!

Example 1:

```
Please enter the first test score: 75
Please enter the second test score: 80
Please enter the third test score: 82
Please enter the fourth test score: 90
Please enter the passing score: 65
Your test scores: 75, 80, 82, 90
The average is 81.75, which is passing.
```

Example 2:

```
Please enter the first test score: 72
Please enter the second test score: 56
Please enter the third test score: 53
Please enter the fourth test score: 74
Please enter the passing score: 65
Your test scores: 72, 56, 53, 74
The average is 63.75, which is NOT passing.
```

Copy your program's code from Replit and paste it below:

(use the **Courier New** font for code-style!)

LAB CONTINUES ON NEXT PAGE!

Programming Challenge 2: Stock Market!

Comment out or delete the code you wrote for the first problem.

Write a program to accept the following three things from the user:

- a stock "ticker symbol" (for example, "GOOGL" or "AAPL" or "GE")
- how many shares of that stock the user owns (assume whole number shares), for example 8 or 125 or 2000
- the change in value of the stock price for that day as a positive or negative decimal (for example, entering 12.75 indicates that the stock *increased* in value by \$12.75 per share, and entering -7.59 indicates the stock *decreased* in value by \$7.59 per share).



After the user enters these values, inform the user how much the *total value* of their stock changed that day, as a number ***rounded to the nearest integer***, and whether that value was gained or lost (remember that you learned rounding strategies yesterday!).

Example: If the user owns 53 shares of Google (GOOGL), and the price changed by 7.52 per share (positive, so an increase), the total value of their stock changed by $53 * 7.52$, which is a gain of **\$398.56**, or **\$399** when rounded to the nearest integer:

```
Enter a stock ticker symbol: GOOGL
How many shares do you own? 53
Enter the change in stock price as a positive or negative decimal: 7.52
Change in stock value: $399 dollars
```

Example: If the user owns 76 shares of Apple (AAPL), and the price *decreased* by 10.37 per share, the total value of their stock changed by $76 * -10.37$, which is **-\$788.12**, or **-\$788** when rounded to the nearest *negative* integer:

```
Enter a stock ticker symbol: AAPL
How many shares do you own? 76
Enter the change in stock price as a positive or negative decimal: -10.37
Change in stock value: $-788 dollars
```

Example: If the user owns 2005 shares of General Electric (GE), and the price *decreased* by 1.97 per share, the total value of their stock changed by $2005 * -1.97$, which is **-\$3949.85**, or **-\$3950** when rounded to the nearest *negative* integer:

```
Enter a stock ticker symbol: GE
How many shares do you own? 2005
Enter the change in stock price as a positive or negative decimal: -1.97
Change in stock value: $-3950 dollars
```

Copy your program's code from Replit and paste it below:

(use the **Courier New** font for code-style!)

--

LAB CONTINUES ON NEXT PAGE!

Programming Challenge 3: Exact Change!

Comment out or delete the code you wrote for the first problem.



Write a program to accept a decimal amount of money from the user, such as 1.37, 5.00, or 2.94 to represent \$1.37, \$5.00, or \$2.94.

Your program should determine what that value would be in *coins* -- quarters, dimes, nickels, and pennies -- and do so *using the fewest coins possible!*

For instance, if the user types in **1.37**, your program should read that as \$1.37, and tell the user that the amount can be represented in a *minimum of 8 coins*: 5 quarters, 1 dime, 0 nickels, 2 pennies:

```
>_ Console x Shell x +
Please enter an amount in dollars and cents: 1.37
The minimum number of coins is: 8
5 quarters
1 dimes
0 nickels
2 pennies
```

An entered amount of **5.00**, representing \$5.00, can be represented with a minimum of 20 coins, all quarters:

```
>_ Console x Shell x +
Please enter an amount in dollars and cents: 5.00
The minimum number of coins is: 20
20 quarters
0 dimes
0 nickels
0 pennies
```

And an entered amount of 2.94, representing \$2.94, can be represented with a minimum of 17 coins: 11 quarters, 1 dime, 1 nickel, and 4 pennies:

```
>_ Console x Shell x +
Please enter an amount in dollars and cents: 2.94
The minimum number of coins is: 17
11 quarters
1 dimes
1 nickels
4 pennies
```

Note that your program should print *all numbers as integers* -- no decimal points.

The following is *not* acceptable!

```
Please enter an amount in dollars and cents: 2.94
The minimum number of coins is: 17.0
11.0 quarters
1.0 dimes
1.0 nickels
4.0 pennies
```

Numbers should **not** print with a ".0"

Copy your program's code from Replit and paste it below:

(use the **Courier New** font for code-style!)

Make sure both partners have the same code for both problems!

Done!

Submit this document in Google Classroom

Turn in

Sample solutions for all three problems will be posted tomorrow.