1.w Peter | Peer Review

Ali and I did the peer review together. We went through every exercise using IntelliJs shared coding feature. All the exercises were functional the way they were, but we still had differences in our approach, especially regarding readability and what types to use for certain variables.

Ex: 2

```
System.out.println("you worked out this week " + (double)timeOfWorkout * 7 / 60
+ " hours and you Burned " + totalBurnedCaloriesDay * 7 +" Calories for this
week" );
```

Ali and I discussed the possibility of including calculations inside the print statement, which makes coding more efficient but hurts readability. While I also prefer doing it inside the print statement most of the time, we did not come to a definitive conclusion about which approach would actually be seen as better from an experienced programmer.

Ex: 4

#Reassigning variables VS initializing new ones
While Ali reused the 2 variables he used to calculate the addition for the multiplication, I made 5 separate variables. We were unsure which method is actually the better one, since Ali was way more efficient, but I had more readable code. We concluded that it depends on what you are working on (in a big team for example, I would prefer readability, while I would like to be efficient on a personal project)

Ex: 7

#Integer.MAX_VALUE;
As in most exercises, both of us solved the issue without much of a problem, but our approach was completely different. While Ali looked up the max amount an Integer can hold in Java, I looked up a method that already saves this value. So while the result remains the same, we agreed that using the method is usually going to be better practice, especially since it is way easier to remember and less prone to errors.

Ex: 8

While Ali and I both understood the problem and solved it, our comments were quite different. We discussed the way we explained it, coming to the conclusion that we meant the same thing using different ways to describe it.

Ex: 9

```
Scanner scanner = new Scanner (System.in);
System.out.println("How old are you? ");
int age = scanner.nextInt();
scanner.nextLine(); // when pressing enter to the age it's avoiding the
System.out.println("What's your name? ");
String name = scanner.nextLine();
System.out.println("Hi, " + name + " You are " + age + " years old.");
```

Ali was already using the scanner.nextLine() catcher perfectly, but was not sure how it actually worked. I was happy to be able to explain it to him the way I understood it.

Something that we did differently over all the exercises was the types we used for our variables. While I always prefer to use double if not otherwise specified, since it is more accurate, Ali used int for every exercise he could use it for. In my personal opinion, the only reason to use int in calculations like this is to make it more readable for the user. This can be achieved by typecasting just the result, which would hurt the accuracy less than actually using int for everything. Also, when specifying int as an scanner.nextInt() variable, you are denying the user the possibility of even entering decimal values without throwing TypeErrors. So while most people seem to use int more often than double, I have yet to understand the advantage of that.