|  |  |  |  |
| --- | --- | --- | --- |
| IST 261, Fall 2019 |  | Student | Ryan Carey |
| PSU Abington |  | Email | Ruc230@psu.edu |
| Phil O’Connell |  | Journal # | 2 |
|  |  | Covering week #(s) | 3/4 |
|  |  | Due | 9/14/19 @ 9 AM |

How requested features were handled...

Most of this code remained unchanged. In fact I was pretty efficient about how the code was changed in order to get the desired behavior. I used a try catch block to validate user input as requested and implemented an ArrayList. I decided to initialize that ArrayList a little differently than we were shown in class, using ArrayList<Integer> in place of List<Integer> because it meant one less import statement. I had already followed camelCase for the most part, so very little if anything had to be changed there. I’m particularly happy that I was able to complete phase 11 with very few lines. The output of all numbers given was just 3 lines, because I used a for loop to cycle through the arraylist and a single statement afterward that ends the program, so there is no need for braces in this case, which is neat.

Issues…

I had no issues with this project except one, which I was able to debug easily on my own. When using the try catch block, I got stuck in an infinite loop for a minute. Thankfully I’ve recently seen this type of error both in examples in class and my own programs. The scanner needed to be flushed of its EOL characters. A Scanner.nextLine() solved the problem immediately.

Minor Tweaks for Best Practice...

One tweak I picked up in class was using String.equals the reverse direction I had been before. Now it doesn’t matter if answer is initialized, because those comparisons start with a string literal I know is definitely initialized. I also explicitly initialized the answer string variable just to be safe.