## Developer Log

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Date* | *Time (roughly)* | *How long did you work on your project?* | *What did you accomplish? What resource(s) did you consult (people or websites) and what did you learn from them?* | *What did you get stuck on (even if you’re still stuck on it)?* | *What are the strengths and weaknesses of the program?* | *What do you want to work on / get done next time?* |
| 30 Nov | 11:50 am | 60 mins | I brainstormed ideas for my project, figured out what I wanted to do, and wrote a lot of my project. I am nearly done with the addition method and started work on my gradeBoundaries method. | I got stuck on the concept of returning values I was making the grade boundaries method. I soon fixed this when I returned a blank string as a placeholder so my project would compile. | Strengths:   * The program can now generate addition tests in the four difficulties.   Weaknesses:   * Subtraction, multiplication and division have not been added. * gradeBoundaries method isn’t done. * The program is very buggy. | * Make difficulty levels for arithmetic (make a method?). * Finish the gradeBoundaries method. * Write subtraction, multiplication, and division methods. |
| 30 Nov | 1:05 pm | 75 mins | I finished a crude version of my resource class. It is a lengthy 350 lines and is very inefficient, but it works! I also had to learn all about what numbers lead to which grade boundaries in the real world. | I had a big issue with trying to cut down on the length of the difficulty part of the method and even tried turning it into its own method. This didn’t work. I’ll try a different approach next time. For now, though, there’s a lot of code that appears four times. That’s very inefficient. | Strengths   * The program includes all essential components and works! All four types of arithmetic (with a combo mode) and all difficulties work!   Weaknesses   * The program is challenging to read and very buggy. The code is very inefficient. | * Create a difficulty method to cut down on code. * Write a crude driver program. |
| 30 Nov | 2:30 pm | 80 mins | I created a test method to combine most the code from addition, subtraction, multiplication, and division methods. I also cut down my code from 350 lines to 184 lines by cutting redundancies and inefficient code. I cut out pointless and buggy for loops in favour of simpler and faster if statements. I Removed pointless arrays. I added a boolean array to replace a bunch of boolean variables I had added previously. I created a driver program. Created a difficulty nested if-for loop statement. | The driver class, when run, would not produce an output, despite no lexical or syntax errors. I will ask Mr. Ayyappan about this. | Strengths   * The program includes all essential components and works! All four types of arithmetic (with a combo mode) and all difficulties work! * The program is much easier to read, and the code is much more efficient.   Weaknesses   * The program is very buggy. | * Troubleshoot driver class. Clean up code in the resource class. * Merge the gradeBoundaries method with the test method to cut down on lines used because there is no functional reason why the gradeBoundaries method has its own method. It’s pointless, confusing, and takes up space. * Will make a test table for the gradeBoundaries method before I get rid of it. This will tell me if code works. * Will make test table for test method. * Troubleshoot the results I get from both of these test tables. * Will make test table for driver class. * Create a while loop that randomly generates numbers until they can be divided with no remainder. (Modulo) * Create a combo mode that does all types of arithmetic at once. (Math.Random) * Make sure for loop doesn’t ask for the type of problem you want every time. (Put for loop inside of arithmetic if statement) |
| 30 Nov | 8:03 pm | 60 mins | I created a combo mode that does all types of arithmetic at once. I made sure the for loop doesn’t ask for the type of problem you want every time (by putting the for loop inside of arithmetic if statement). I created a while loop that randomly generates numbers until they can be divided and create a 0 (remainder) (modulo) inside of the division method. I made it the question generator could generate so negative numbers. Now, I am at 239 lines of code. | The program crashed, and then I lost 30 minutes of work. Other than that, it was smooth sailing. The driver class, when run, would still not produce an output. I think it has something to do with the arithmetic methods. | Strengths   * The program includes all essential components and works! All 4 types of arithmetic (with a combo mode) and all difficulties work! * The program is much easier to read, and the code is much more efficient.   Weaknesses   * The program is very buggy. | * Troubleshoot driver class. Clean up code in the resource class. * Merge the gradeBoundaries method with the test method to cut down on lines used because there is no functional reason why the gradeBoundaries method has its own method. It’s pointless, confusing, and takes up space. * Will make a test table for the gradeBoundaries method before I get rid of it. It will tell me if code works. * Will make test table for test method. * Troubleshoot the results I get from both of these test tables. * Will make test table for driver class. |
| 2 Dec | 9:20 am | 80 mins | I fixed the driver class so it would run. It was an issue I had with running a for loop inside of the resource class. To fix an issue I had with it asking me the same question, I had to transfer the random number generation to the inside of the arithmetic methods. This ended up working. I made truth tables for the error. I worked with some test tables to troubleshoot the test method and the driver class. Finally, I made a test table for gradeBoundaries. | I spent 50-60 minutes trying to fix the driver class. It ended up working. The issue was with one < instead of a >. Afterwards, I dealt with the issue that it would keep asking me the same question over and over when running the program.  These two things caused me major issues this session. | Strengths   * The program includes all essential components and works! All four types of arithmetic (with a combo mode) and all difficulties work! * The program is much easier to read, and the code is much more efficient.   Weaknesses   * The program is very buggy. | * Merge the gradeBoundaries method with the test method to cut down on lines used because there is no functional reason why the gradeBoundaries method has its own method. It’s pointless, confusing, and takes up space. * Will make a test table for the gradeBoundaries method before I get rid of it. It will tell me if code works. * Troubleshoot the results I get from both of these test tables. * Will make a test table for driver class. |
| 2 Dec | 11:35 am | 170 mins | I Troubleshot the driver class. I Merged the gradeBoundaries method with the test method. I Made a test table for the gradeBoundaries method and driver program. I Troubleshot the results I got from both these test tables. I made it so impossible grades could not be awarded. After all was said and done, my project is 100% complete. | I was having issues with the calculation of the grade of the ‘player’ of the StudyBuddy. It turns out it was an issue that existed with casting, or as a matter of fact, the lack thereof. I was trying to calculate a decimal with two integers. This took about an hour to fix. | Strengths   * The program includes all essential components and works! All four types of arithmetic (with a combo mode) and all difficulties work! * The program is much easier to read and the code is much more efficient.   The program is done!!!!! | N/A |