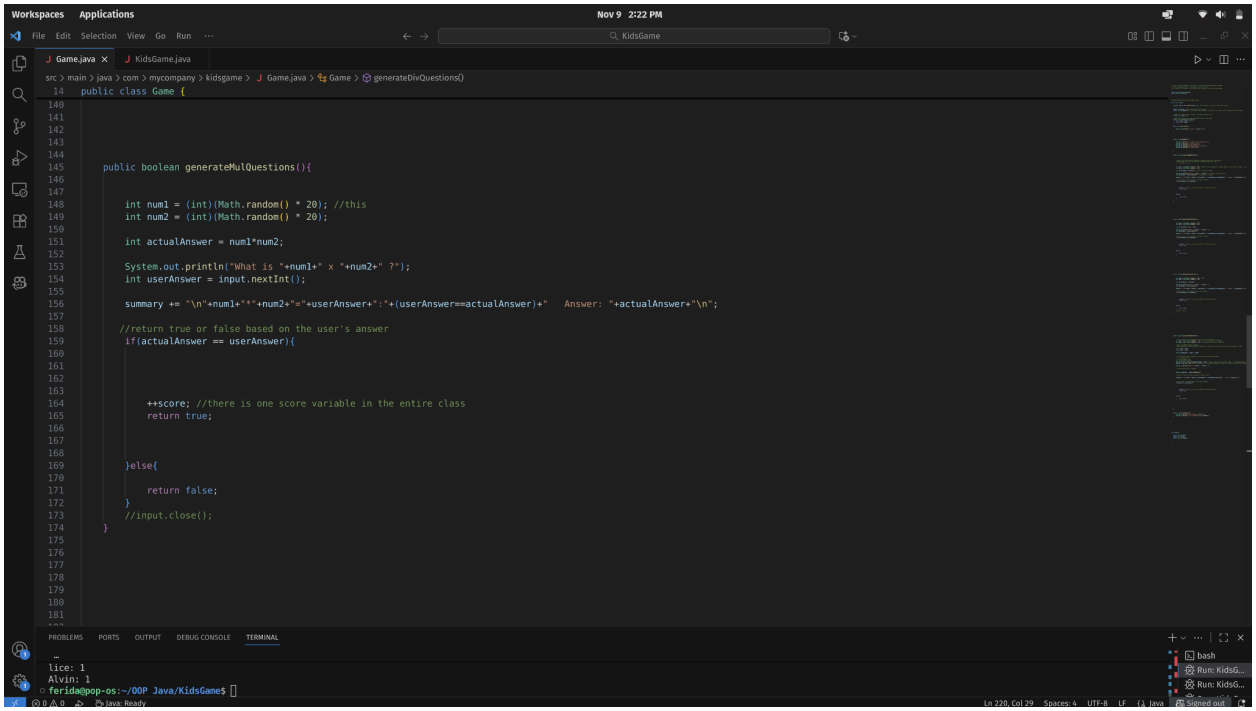
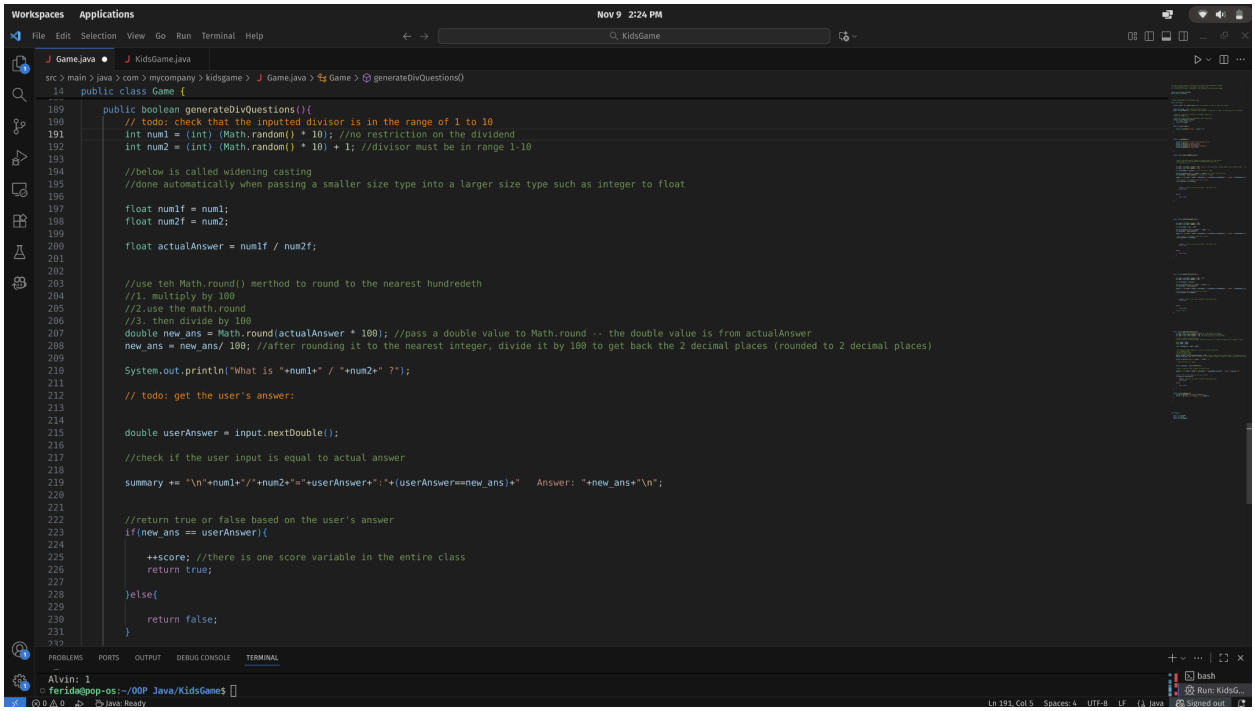
A1:

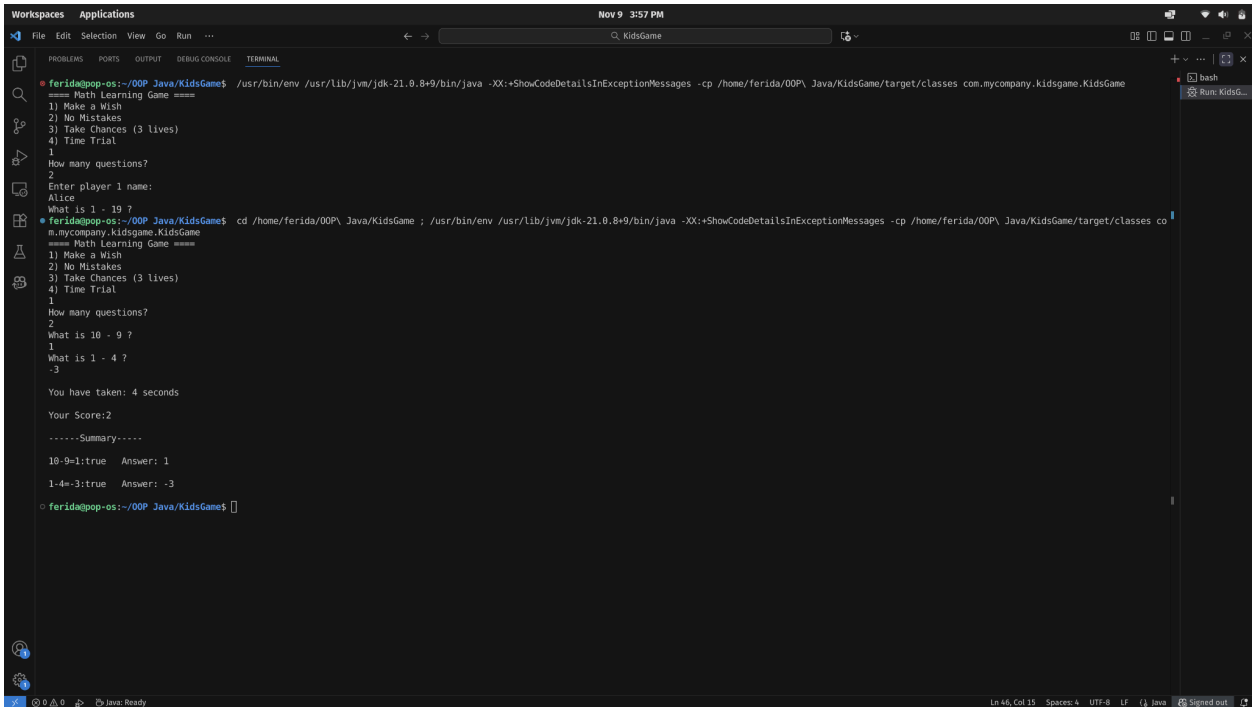


Above is the method generateMulQuestions() method. To implement it, I first created two random numbers using the Math.random() method. Math.random() generates a random number between 0.0 and 0.1, then I multiplied it by 20 to make it between 0-19 (the 0.1 is not inclusive). I used the scanner object in the Game class file to get the user's answer.
Then within the summary, I showed the results of the user, and I used userAnswer == actualAnswer to get a boolean True or False to add to the summary string.
This method returns a boolean value, because it provides the exit condition for some of the while loops in KidsGame file. (Look at cases where I set the variable 'result' = EachArithmeticMethod())



Above is the generateDivQuestion() method. I generated random numbers the same way I did for the generateMulQuestions() method. However, for the divisor, the number had to be between 1-10 to avoid dividing by zero. To fix this, I did (Math.random() * 10) + 1 so that the range would become 1-10 instead of 0-9 (0.0 - 0.1(non inclusive), is what Math.random() usually returns).
The reason I type casted it to int is so that when I print the question to the user, it's easy to read.
Then I got the actualAnswer by dividing the two floats num1f and num2f.
(They're the double equivalents to the random ints I got).
Within Math.round(), I passed the actualAnswer (a float) times 100 to keep the two decimal place numbers for my final answer.
Passing this to Math.round() would round it to the nearest integer.
Then the result I got from this, (new_ans) was divided by 100, so that the two decimal places return to being decimals (after the *100).
Since we used round(), the two decimal places are rounded to the *nearest* hundredth.
Take the nextDouble() from the user

A2

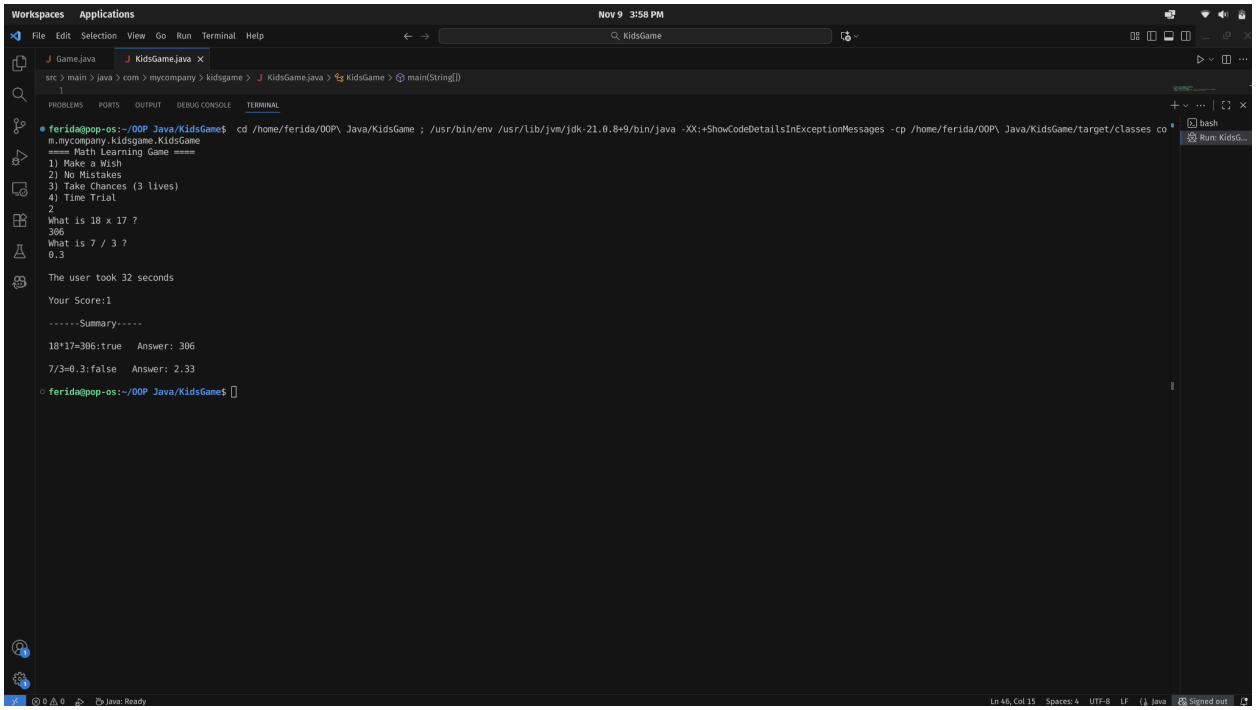


The above screenshot displays a single user's game with Mode 1. Please ignore the first run of my program in the screenshot; look at the second one.

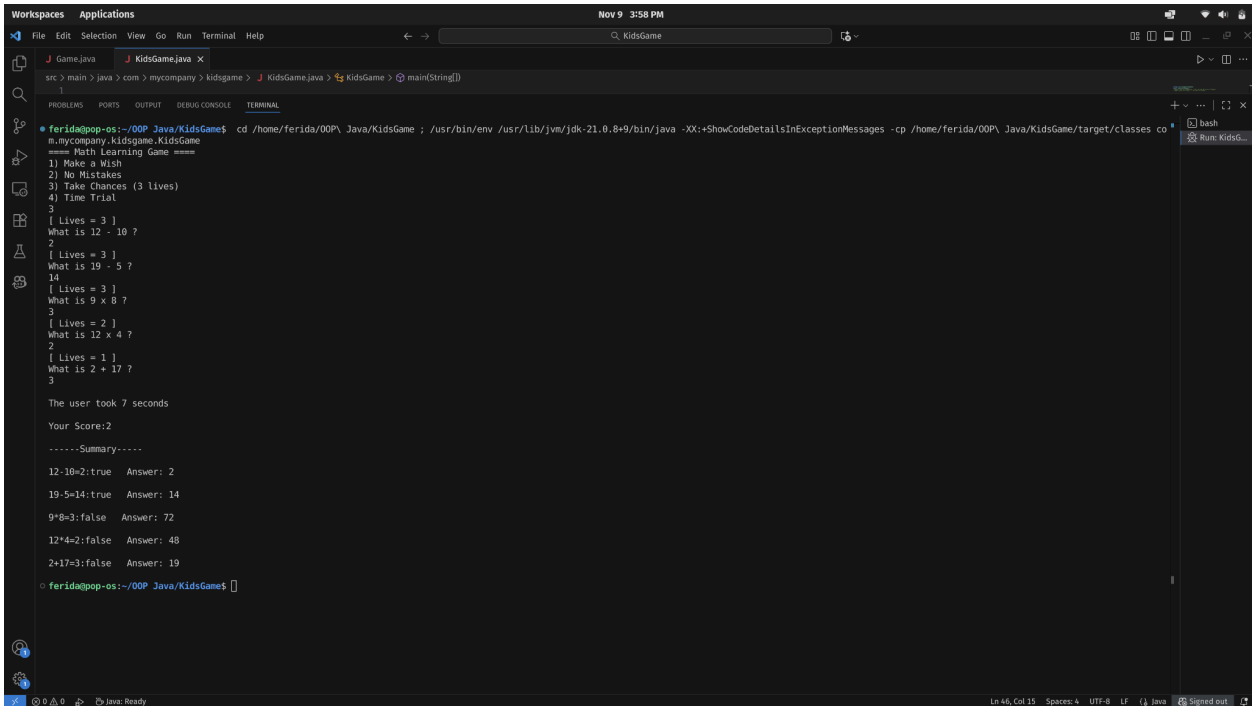
In this Mode, I had a switch case, where each arithmetic function is employed based on a random number generator from 1-4. If the number is 4, it employs the fourth arithmetic method in the fourth case.

To keep track of scores, I increase the game class variable 'score' by one, in each arithmetic method of the game class (only if the user got the answer right).

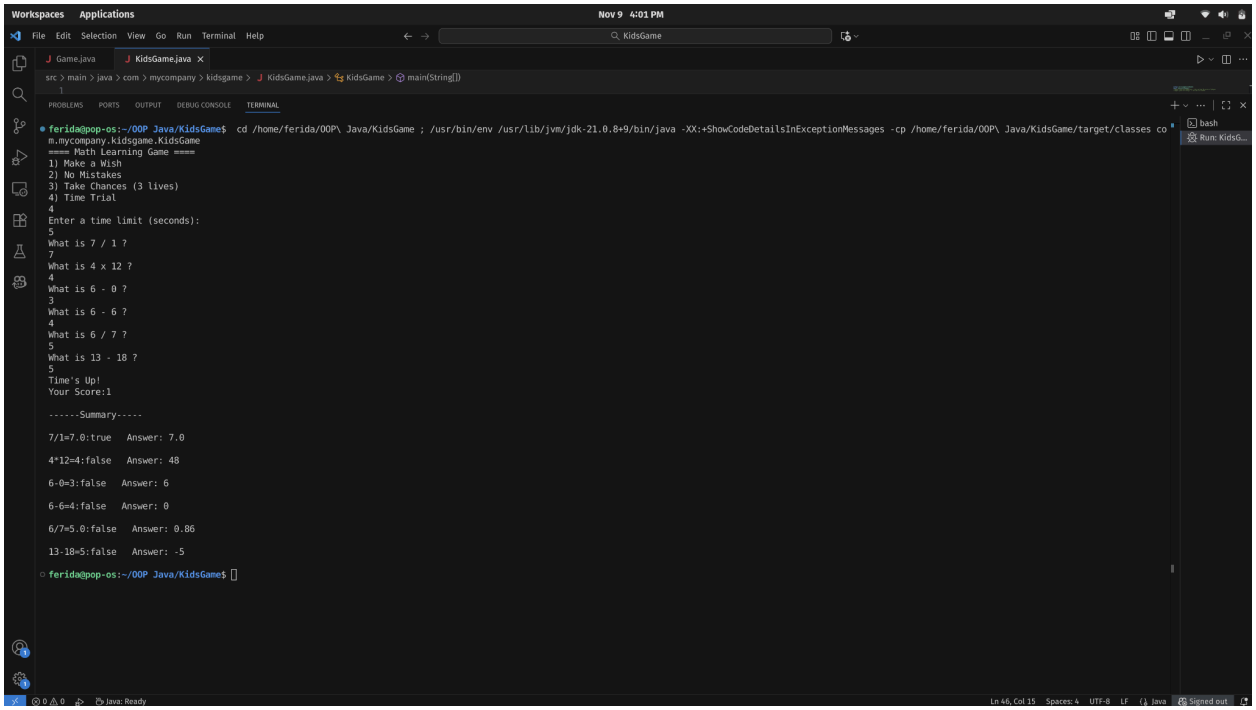
That is why I have an if block in each arithmetic method to check whether userAnswer == actualAnswer or not.



In mode 2 above, to implement the ‘No Mistakes’ feature, I use a switch case to check the return value of each arithmetic function.
Each arithmetic function returns a boolean value, and within the switch case that employs each method (in main, based on a random number generator), I store the return value of the method in a variable called ‘result’. Result is obtained from each method based on the if-else block for (actualAnswer == userAnswer).
Then I use boolean ‘result’ to determine the condition of the while loop that this mode 2 is in. If result == False, mode_is_on = False. This exits the while loop, and then prints the results.



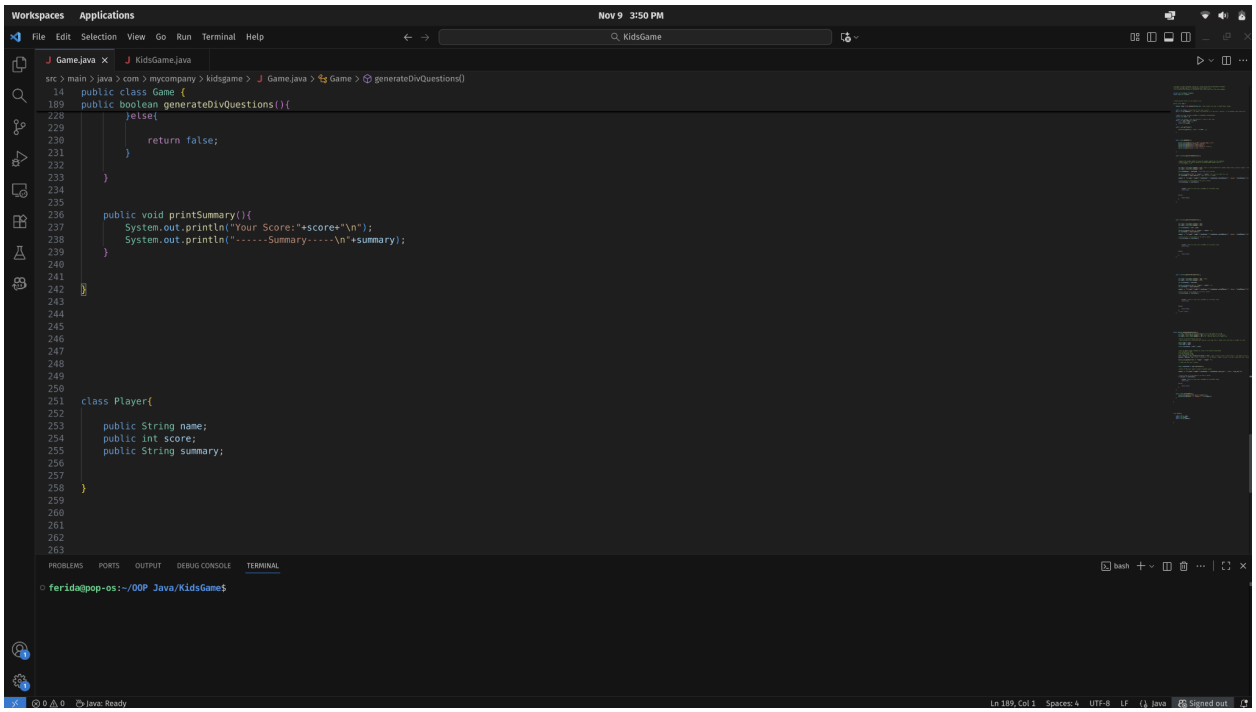
For mode 3 above (single user), I used the concept of encapsulation. Within my game class, I have a private variable called lives which is set to 3. Then I have defined a setter class called set_lives(userInput), and a getter called get_lives.
In my main code, I have declared a variable called lives_left, which is passed into the set_lives method, each time the user gets the wrong answer. (Wrong answer is determined by the switch cases of each arithmetic function explained above).
I use get_lives to print the [Lives = #] that is shown in the screenshot, in each iteration.
Then at the bottom of the while(mode_is_on) loop, I use an if block to determine how many lives are left (using the local variable I have declared in my main program).
if(lives == 0), then mode_is_on == false



In mode 4 above, I had to implement the time limit feature.
To do this, I asked the user for a time limit in seconds, and I read it using the input.nextInt().
Then I used the currentTimeMillis() method to get the current time of the system (which gets it in milliseconds). This was to be the startTime of the while loop.
The condition of the while loop was (the current time) < (startTime + timeLimit)
The current time was different from the startTime of the while loop, because it was calculated at the beginning of each iteration.
If the startTime + timeLimit was still greater than the current time, that means that the user still gets to be asked a question.
If the limit was reached (current system time was greater than or equal to the startTime + limit), then the while loop would be exited, and the results would be displayed.

*NOTE: The scoring is all the same for all the modes; I have one class instance variable called ‘score’, which is incremented depending on the user’s correct answer within each arithmetic method in the game class.

B1



In the player class, I added attributes for each player's name, score, and summary of their results. I did not use the summary of results anywhere for printing to screen, because it was only a requirement to have it included in the class itself.
I did not explicitly create a constructor, instead, each time I got an answer from the user regarding the player's name, I used: (player.name = input.nextLine())
For the score, I would append it to the respective player object using the same format described above. The same idea for the summary.

The sequence of code for competition loop:

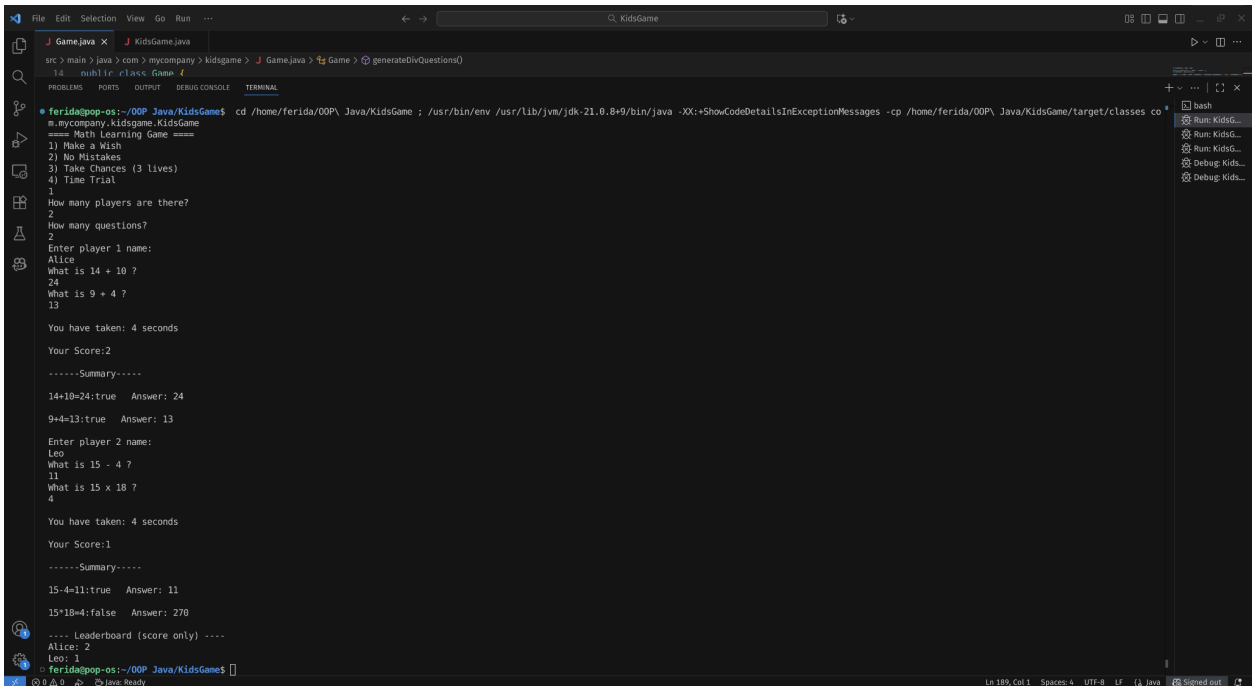
1. I would ask the user for how many players, and store it in a variable called 'players'. Before this, I asked them what mode they wanted to play, stored in 'mode'.
2. Then I had one large if-else-else-else block to check whether user chose mode = 1, 2,3,or 4.
3. In mode 1, I ask user for how many questions, stored in 'nQuestions'
4. Using the 'players' variable, I used a while (i<=players) → int i = 1. Within the while loop, I had a for loop to iterate through nQuestions to ask user.
5. Before the for loop, I would create a new player object, and ask for their name, and reset the game object's summary, and score class variables (so that each user starts from 0).
6. I had created an array of player objects before entering the if-else-else-else block, using the 'players' variable to determine its size

7. So at the end of each while (i <= players) iteration, I would append arr_players[i - 1] = player. (the current iteration player object, with i-1 for indexing)
8. For the leaderboard, I used the Arrays.list(Comparator.comparingInt()).reversed, to sort through the scores of each player object. Then I used reversed() to order the array from highest to lowest score. These methods were present in the Professor's Arrays in 2D slides.
9. To print the leaderboard, I went through this array in a for loop, to access each score, and print it along with the player's name. This step was done at the very end of the program.

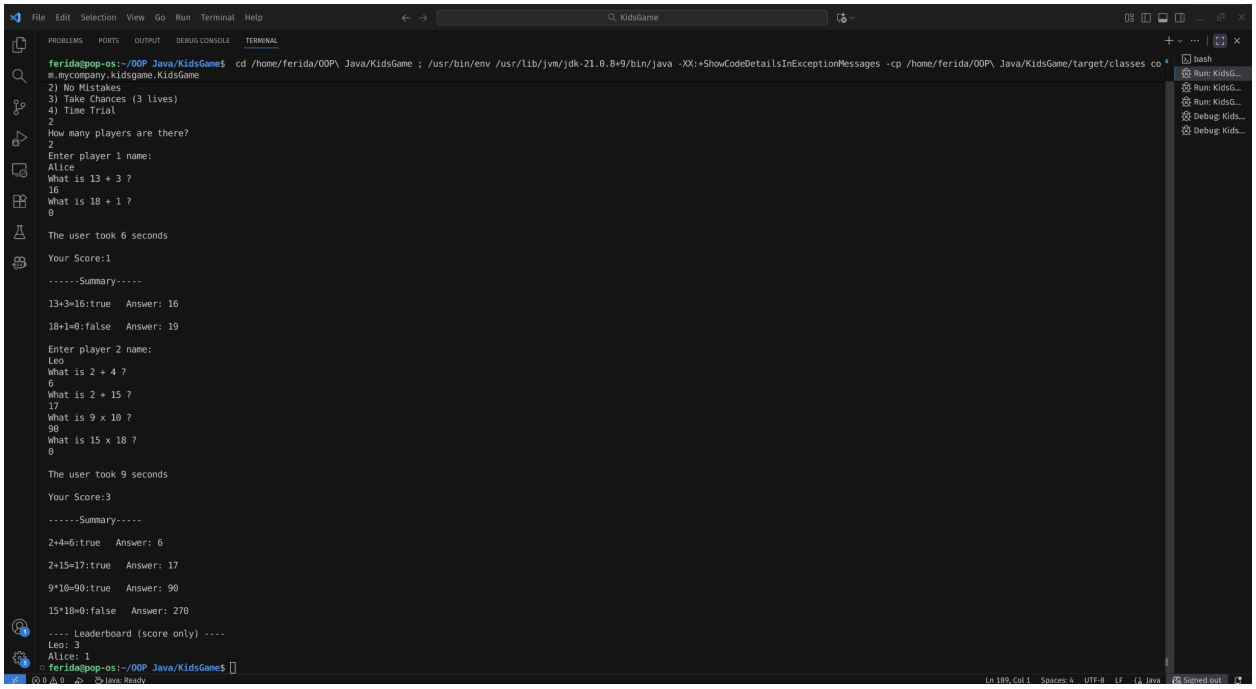
→ These steps are repeated for the rest of the modes. The other modes are in else blocks, with their respective functions/features. However the resetting score, and summary, and while loop condition are the same for all modes for the multiplayer version.

B2

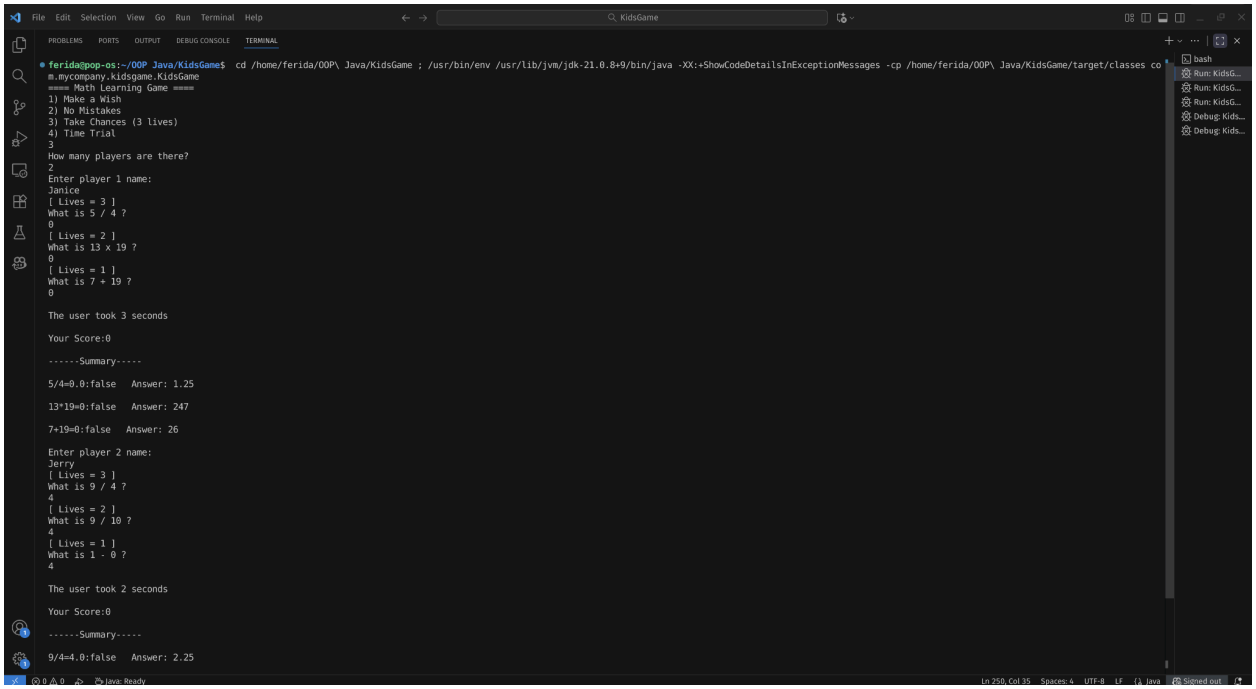
Mode 1:



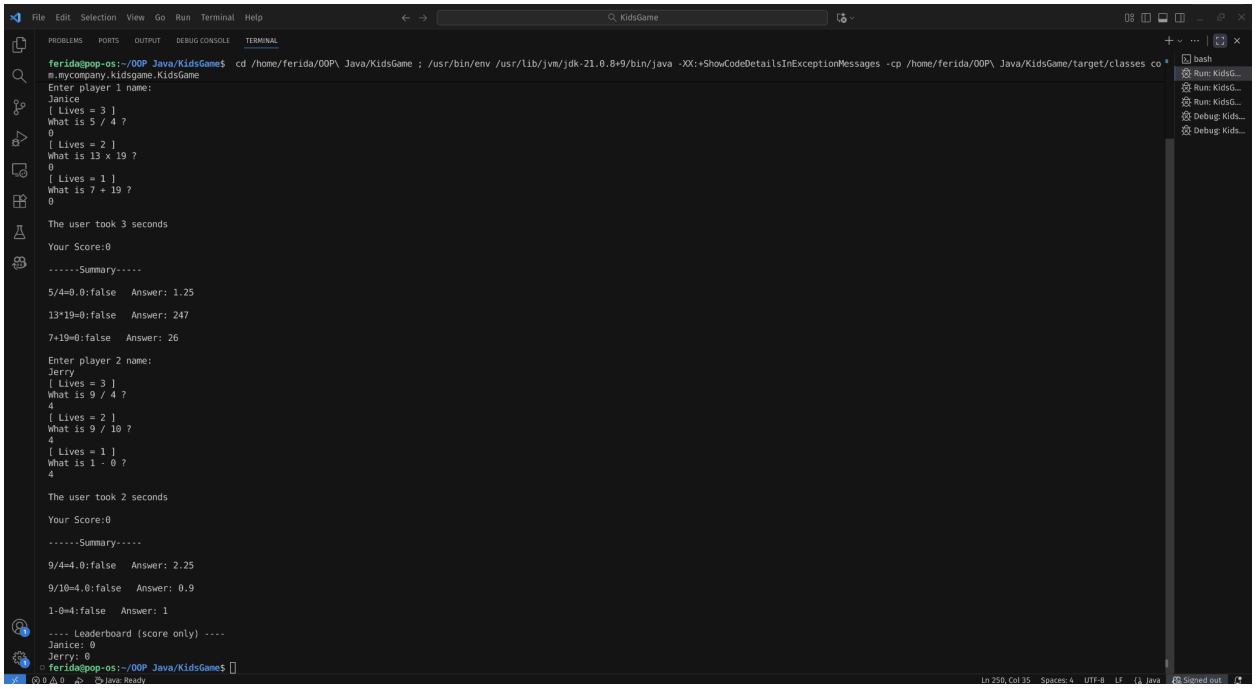
Mode 2:



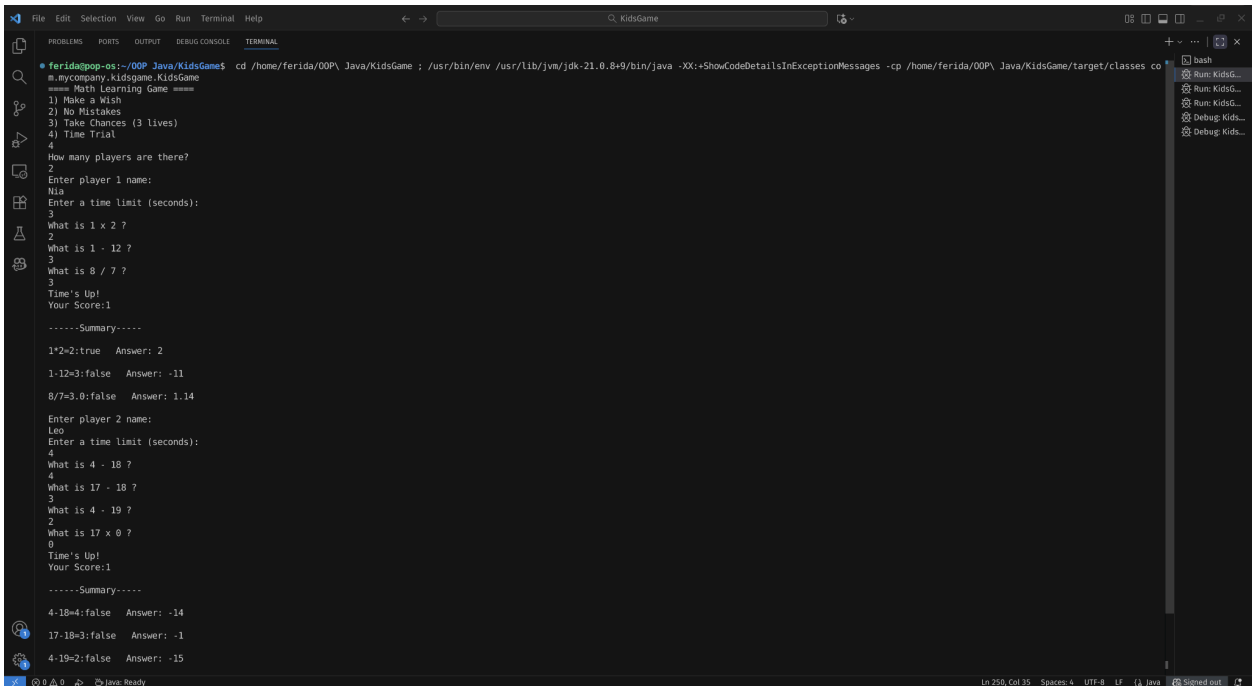
Mode 3-P1 (Leaderboard couldn't fit in screenshot):



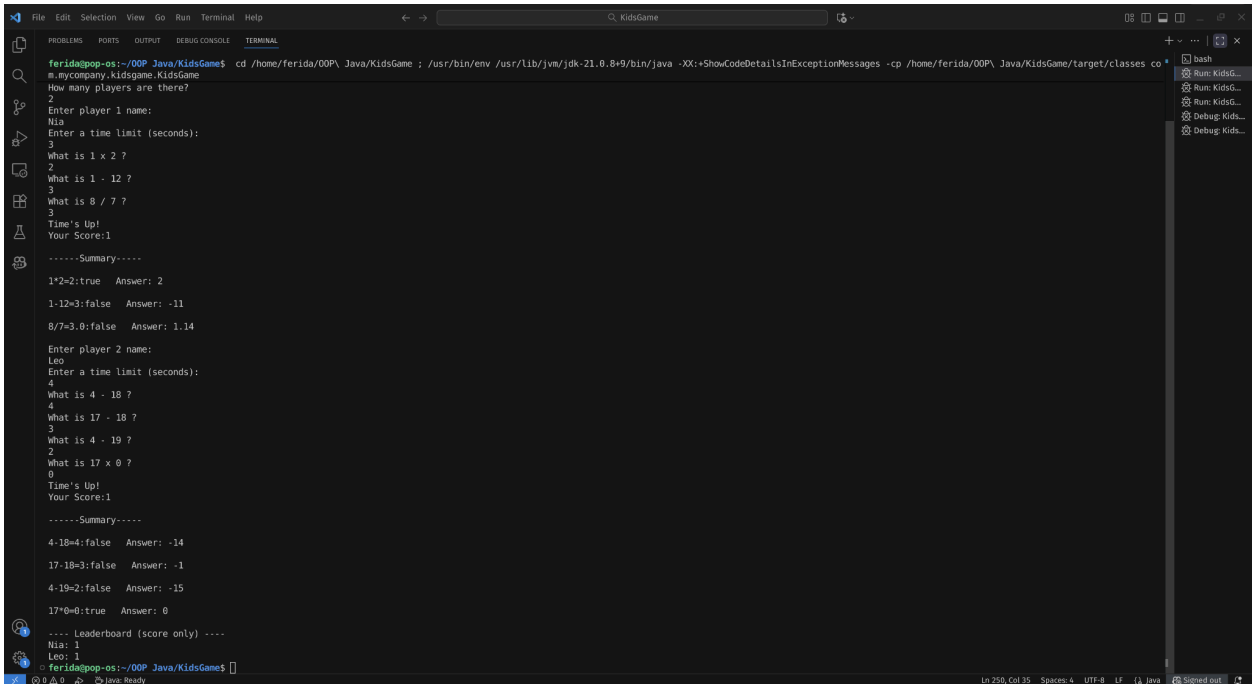
Mode 3-P2 (With leaderboard)



Mode 4-P1:



Mode 4-P2:


File Edit Selection View Go Run Terminal Help
KidsGame
PROBLEMS PORTS OUTPUT DEBUG CONSOLE TERMINAL
ferida@pop-os:~/OOP Java/KidsGame$ cd /home/ferida/OOP\ Java/KidsGame ; /usr/bin/env /usr/lib/jvm/jdk-21.0.8+9/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/ferida/OOP\ Java/KidsGame/target/classes com.mycompany.kidsgame.KidsGame
How many players are there?
2
Enter player 1 name:
Nia
Enter a time limit (seconds):
3
What is 1 x 2 ?
2
What is 1 - 12 ?
3
What is 8 / 7 ?
3
Time's Up!
Your Score:1
------Summary-----
1*2=2:true Answer: 2
1-12=3:false Answer: -11
8/7=3.0:false Answer: 1.14
Enter player 2 name:
Leo
Enter a time limit (seconds):
4
What is 4 - 18 ?
4
What is 17 - 18 ?
3
What is 4 - 19 ?
2
What is 17 x 0 ?
0
Time's Up!
Your Score:1
------Summary-----
4-18=4:false Answer: -14
17-18=3:false Answer: -1
4-19=2:false Answer: -15
17*0=0:true Answer: 0
---- Leaderboard (score only) ----
Nia: 1
Leo: 1
ferida@pop-os:~/OOP Java/KidsGame$
bash
Run: KidsG...
Run: KidsG...
Run: KidsG...
Debug: Kids...
Debug: Kids...
Java: Ready
Ln 250, Col 35 Spaces: 4 UTF-8 LF Java Signed out