My program demonstrates a single round of the game farkle, in which you are given 6 dice and are ordered to make melds out of them, which are specific combinations of dice that give points to the player. If there are no points possible, your turn is over and you are awarded no points. I implement this through 3 classes: the Die, Meld, and Combos class. Within main I initialize each class so I can use the methods within them, and start by rolling 6 dice for the user. It then checks for a farkle and ends the program if the score is 0, after resetting the score, we enter a while loop that asks for the position of the dice you'd like to add to the meld ArrayList. When a dice is added to the meld array, a 0 is replaced in the users hand, and the dice is added to a meld array list at the index of the die value; for example, if I input two 5s, the array would have a 2 at index 5. Once the user is satisfied with their inputs, they input -1 to finish the meld, which then goes through the different combos within the Combos class, and the score is calculated.

I was having issues testing the code for specific cases, as I hadn't made a test file so I ran the program until I got different combinations; with more time I would've constructed a test file that would make my life much easier. I was also getting confused with myself with the amount of objects I was using in the code, as I was using ArrayLists which have a very different syntax than arrays when manipulating them, it was also difficult for me to convert the objects into integers. I fixed this by making an integer ArrayList and grabbing the sideUp() from each die to convert to integer. If I had more time, I would convert the objects to integers inside the main function to make crossing over easier, as it was also difficult for me to return the unused dice in the melds arraylist back into the user's hand once the melds are calculated.


Die
-sideUp:int
-numSides:int
-DEFAULT_NUM_SIDES;int
-DEFAULT_SIDE_UP:int
+ Die()
+ Die(numSides: int)
+ Die(numSides: Integer, startingSide: Integer)
+ roll(): void
+ getSideUp(): Integer
+ getNumSides(): Integer
+ toString(): String
+ compareTo(otherDie: Die): int
Meld
- melds: ArrayList<Die>
- diceCount: ArrayList<int>
- combo: Combos
- score: int
+ Meld()
+ addToMeld(die: Die): void
+ checkMeld(): void
+ combineDice(): void
+ displayMeld(diceCount: ArrayList<Integer>): void
+ addLeftoverDice(dice: ArrayList<Die>): void
+ isFarkle(dice: ArrayList<Die>): boolean
+ size(): Integer
+ getScore(): Integer
+ setScore(newScore: int): void
combos
ArrayList<Die>
combos
- farkle: boolean
- score: int
+ Combos()
+ triples(diceCount: ArrayList<Integer>): void
+ threePairs(diceCount: ArrayList<Integer>): void
+ straight(diceCount: ArrayList<Integer>): void
+ singleOnes(diceCount: ArrayList<Integer>): void
+ singleFives(diceCount: ArrayList<Integer>): void
+ check(diceCount: ArrayList<Integer>): void
+ displayHand(diceCount: ArrayList<Integer>): void
+ setScore(newScore: int): void
+ getScore(): int