Cody Kimmins

Project 1 Report

For the beginning of Project 1 I started with the Mean, Median, and Mode program. With the MMM program I first had to refresh my knowledge of how each formula works and is used. Working out mean and median were straightforward, although the implementation of mode was tricky because of the multiple loops looking through the array for the most frequent number. The Set Operations program uses 3 different array lists for universal, A, and B. The Union method I created loops to go through sets A and B having an if else statement to only add if set A is not in B and vice versa. Intersection was simple because this is just adding both sets having every number included. The complement method was tricky but researching the logic helped me. The complement of A is comparing it to the universal set so when adding to the new list it is elements that are not in A but in universal set.

The Combination and Permutations program I had to create a method for the factorial and use that within the permutation and combinations methods. The most challenging part with this was having to use BigInteger since the values I was testing were too big for int. The book problems I tested this with have very large factorials and was at first getting a negative value or zero. I had to research this and found BigInteger which essentially allows for very big calculations. To multiply or divide I had to store each result into a variable and then apply the appropriate mathematical operation. I used the combination and factorial methods in my Standard Deviation and Variance programs since they are relevant in their formulas. Although I tried to import the classes from the other project, it was confusing to implement with my IDE (IntelliJ). My solution was to just copy over the methods and redo them to make it int instead of BigInteger since values that big are not needed.

The next few programs I worked on were the Monty Hall and Birthday Paradox programs. I had extensive research on the Monty Hall problem and studied the logic of how it works including the correct result. I created two main methods called StayWins and SwitchWins. StayWins loops x number of times and checks if two random numbers are the same. SwitchWins is a little more complicated and requires an increased chance of the two random numbers to be the same. Essentially the chances between picking 1 out of 3, it would then be 1 out of 2. The Birthday program was also challenging to me because of the concept. The program has a Person class to create and store birthdays as well as a method to generate random birthdays. The method to run the trials first generates a random birthday, then loops through the arraylist and checks if there a match. If there is a match, it adds to birthdayMatches and keeps looping until number of trials is met. The result outputs the user’s input, birthday matches, and percentage. For a 50% match there must be 23 people and inputting that into my program, the result is ~50%.

The plotter, salter, and smoother program provided a unique challenge as I had to first learn about csv files as well as how to print to one. For this program I used the slope formula because I had a hard time using readers and writers for the csv file. In the program I first generated 15 X values and Y values that multiplied by 2 and added 3 to get the straight line. Salting the values Salting the values loops through and reads the file while replacing the values with new random ones. The smoother method attempts to reverse the random change of numbers by using the salted values and dividing to “smooth” the data set.