The program begins with creating an array with the assigned variables, since they were all numbers, the class of the array is int.

I then created some more arrays that I would use to store the duplicates in later on in the program. I knew that anything going into them would also be an integer, so this again was the class of the array. Since array length cannot be altered, I set the length of the array equal to the length of my intial array. Since we were looking for duplicates, it is not possible for these second arrays to be larger, it I think it is ok if they are not full, having some null values.

I then created a for loop that would take one value at a time and compare each other value to it, checking if they were the same. This would find duplicated. If they were duplicated, it was sent to the variable track.

I then stored these duplicated in the array made earlier called “dupeTrack”, within this loop, also checking if duplicates had been found already. If the duplicate had been accounted for already, the loop would break, then continue to the next step.

I then did another loop that would go through again, this time counting how many times each duplicate appeared. Once again using the variable “now” as the current integer that was being compared.

Assuming that there were duplicated within the array, it then created an “if” statement that would store how many duplicates that were found in “dupes” array and the number of times in which they appeared in “counts” array.

The last part of the program was a final if else statement. If “dupeTrack”=0, this meant there was no duplicates in the array and it would print to the user to tell them this, ending the program.

If there were duplicates, this would trigger the “else” part. This is made up of a for loop, that would go through each duplicate found and print it out. I used “a” to loop through, printing from the array “dupes” that prints the integer that was appearing, and “counts” that would print how many times said integer appeared.