Unit 3 – Worksheet 03 More with Arrays Name: _____

#1-7 If an array is correctly initialized, write the final array (row-by-column, including the array elements) once all of the code has been executed. If not correctly initialized, state that the array is not appropriately initialized.

1. *int[ ][ ] theArray = {1, 2, 3, 4};*

2. *int[ ] theArray = new int[5];*
   theArray[1] = 2;
   theArray[4] = 8;

3. *int[ ][ ] theArray = new int[3][2];*
   for(int i=0 ; i<theArray.length ; i++)
   for(int j=0 ; j<theArray[i].length ; j++)
   *theArray[i][j] = (2*i)*(j+1);*

4. *double[ ] theArray = {2, 3.5, 6.4};*

5. *String[ ][ ] the Array = { "a", "b", "c" },*
   {"d", "e", "f" };

6. int[ ][ ] theArray = { {1,2,3},
   {4,5,6} };
   for(int i=0 ; i<theArray.length ; i++)
   theArray[i][0]=9;

7. *int i=0;*
   int[ ] theArray = {5,6,9,1,-3,10,-2};

   while(theArray[i]>0) {
   theArray[i]+=3;
   i++;
   }

8. If theArray created in problem #4 is used, what is the output for the following code?
   double answer=0;
   for(double stuff : theArray)
   answer+=stuff;
   System.out.print(answer);

9. If theArray created in problem #6 is used (the original array), what is the output for the following code?
   int answer=0;
   for(int[] row : theArray)
   for(int element : row)
   *answer=element*2+answer;*

   System.out.println(answer);

#10-13 Use the following code to determine the output for each problem:

```
String[] taco = new String[3];
taco[0]="Soft";
taco[1]="Hard";
taco[2]="Salad";
```

10. *System.out.print("I ordered 10 "+taco[3]+" tacos.");*

11. *System.out.print("My favorite is "+taco[0]+" tacos!");*

12.
```
String hold = taco[1];
taco[1]=taco[2];
taco[2]=hold;
System.out.print("Types of tacos: ");
for(int i=0 ; i<taco.length ; i++)
    System.out.print(taco[i]+" ");
```

13. *Be sure to notice what really happened in problem #12. This process will also be used when learning how to sort and search data using JAVA. What did the code do and how did it do it? Explain (write sentences!).*

14. **Review:** Create a new class called **pairsOfShoes**.

- Create a set of parallel Arrays called **name** and **shoes** (String Array to store names and an integer Array to store number of shoes).
- Ask the user for how many people are in their family and store their entry as a variable. Use a for-loop to collect the names and the number of pairs of shoes owned for each member of the users family. The names should be stored into the name Array while the number of pairs of shoes owned should be stored into the shoes Array.
- At this point, you will have created parallel arrays … remember this just means that you have created two separate arrays with elements of the same index numbers having significant meaning (they are "connected"). For example, name[2] and shoes[2] are directly connected as they tell us how many shoes a specific individual owns.
- Write a second for-loop that will output the data about the user's family in a neatly organized table. Inside this for-loop you should write code that will keep a running total of how many total pairs of shoes are owned in the family.
- Write one final print statement that tells the user how many total pairs of shoes are owned in the family.

If done correctly, the following is an example of what the output might look like after data is collected:

```
Here are the results for your family:
Julie       5
Matt        6
Emilie      10
Zach        2
Elijah      5
Your family owns a total of 28 pairs!
```