CS 125 - Lecture 14

**Objectives:**

Objectives: Processing 1D arrays
MP3 - due in three days.

The following activities reinforce the readings....

1. **True/False?**

Q0. A Java array is an object. That is, a declared array variable does not actually hold the array, it refers to the array instead.

Q1. A Java array can hold a mixture of primitive types (e.g., integer in cell 0, boolean in cell 1, double value in cell 2, etc.)

Q2. The cells (or 'entries' or 'elements') of an array are indexed by an integer.

Q3. The first cell of an array is at index 1. To add 10 to the first score `scores[1] += 10;`

Q4. `scores.length = 500;` changes the size of the array.

Q5. The last cell in the array 'scores' will be `scores[scores.length -1]`

Q6. `new int[] {3,5,6,10}` creates an integer array of length 4.

Q7. `new int[99999]` creates a large integer array, each cell is initialized to zero.

Q8. `new char[50]` creates a character array with 50 cells, each cell is initialized to a space.

Q9. `new String[50]` creates a String array with each cell initialized to an empty string.

Q10. `scores[-1]` or `scores[scores.length]` will produce IndexOutOfBoundsException.

4. **Does `scores[0]` change in the following code? Why?**

```
int[] scores = readScores();
String name[]= readNames();
int[] b = scores;
b[0] = 0;

TextIO.putln( name[0] + " : " + scores[0]);
```

2. **The bank account swindler:**

```
double[] cash = new double[] {-33.,102.,515.,10004.,42.07,…};
for(int i=0; i<cash.length; i++) {
   if (cash[myAccnt] < cash[i] ) {
      // Please Fix - It should swap the values.


      cash[myAccnt] = cash[i];


   } //if
} //for
```

3. **Modify the code below to roll three dice.** It should keep rolling until the dice values are unique. You'll need to i) create a new variable (dice3); ii) roll dice 3; iii) change the foundSolution expression and iv) the return expression should represent the number of iterations required.

```
/** Rolls three simulated 6 sided dice until all die values
/*  are unique.
public static int rollThreeUniqueDice() {
   int dice1 = 0, dice2 = 0;

   boolean foundSolution = false;
   while (!foundSolution) {
      dice1 = 1 + (int) (Math.random() * 6.0);
      dice2 = 1 + (int)( Math.random() * 6.0);









   }


}
```

5. **What will be the final contents of the array?**

```
int [] numbers = new int[] {10,11,12,13};

for(int i=0; i<numbers.length; i++)

   numbers[i] = numbers[ numbers.length -1 - i];
```

6. **What is meant by a partially full array?**

8. Complete the following method to return the **array index** of the smallest value. Do not print anything out.

```
int findIndexOfMinimum(int[] array) {
   int smallest = 0; // index of smallest
   for(int i=0; i< array.length; i++) {

     if ( array[i] _____________ ) smallest= ________;

   }

   return _______________;

}
```

9. **Why is the following algorithm called selection sort?**

(The method "findIndexOfMinimum" starts the search from index 'i' not index 0)

```
for(int i=0; i<array.length;i++) {
    int smallest = findIndexOfMinimum(array, i);
    swap(array,i,smallest);
}
```

http://en.wikipedia.org/wiki/Selection_sort#mediaviewer/File:Selection-Sort-Animation.gif

11. **What will be the final contents of 'myarray'?**

```
String mesg = "Vewol Swap";
char[] myarray = mesg.toCharArray();

for(int i=0;i< myarray.length; i++) {
  if( myarray[i] =='o') myarray[i]='e';
  if( myarray[i] =='e') myarray[i]='o';
}
```

7. **Carefully execute the following code by hand and note the variables values as they change.** (i) Determine the final value of each variable. (ii) Determine what the code does.

| i: | j: | count: |
|---|---|---|
| result: | | |

```
int[] arr1 = {10, 20, 30, 40}; //sorted values
int[] arr2 = {18, 20, 25, 99}; //sorted values
int[] result = new int[arr1.length];
int i=0,j=0,count=0;
while (i<arr1.length) {
   if(arr2[j] < arr1[i]) j++;
   else if(arr2[j] == arr1[i]) i++;
   else {  // must be true that arr2[j] > arr[i]
      result[count] = arr1[i];
      i++; count++;
   }
}
```

10. **PARALLEL ARRAYS:** Complete this code to print up to 50 movie titles of movies that grossed over $5 million. Print the array index of the highest grossing movie.

```
public static void main(String[] args) {
double[] gross = ... //gross[i] movie earnings of i^th movie (in $m)
String[] title = ... //title[i] movie title of i^th movie.
```

7. **You have a list of favorite movies, String[] movies. Write code to ask for another movie and append it to the end:**

8. **Explain why the following do not make copies:**

```
String s1 = "Hello!"
String s2 = s1;

int[] A = new int[] {101,102,103};
int[] B = A;
```

And explain why the following do not compare the values of the array or string objects:

```
// code continues from above

s2 = "Hello" + "!";
B = new int[] {101,102,103};

if(A == B && s1 == s2) TextIO.putln("Same!");
```

9. **What will be the final contents of 'myarray'?**

```
String mesg = "Vewol Swap";
char[] myarray = mesg.toCharArray();

for(int i=0;i< myarray.length; i++) {
  if( myarray[i] =='o') myarray[i]='e';
  if( myarray[i] =='e') myarray[i]='o';
}
```

3. **Solving "Knight and Knaves" Logic Problems Computer Science Style!**

- Person 1 says "Person 2 is lying"
- Person 2 says "There are two liars here"

```
// 0 = liar, 1 = tells the truth

for (int person1 = 0; person1 < 2; person1++)
   for (int person2 = 0; person2 < 2; person2++){
      // Person 1 says "Person 2 is lying"
      boolean testimony1IsTruthful = person2 == 0;
      // Person 2: "There are two liars here"
      boolean testimony2IsTruthful = person1 + person2 == 0;

      boolean assertion1 = (person1 == 1 && testimony1IsTruthful)
                        || (person1 == 0 && !testimony1IsTruthful);
      boolean assertion2 = (person2 == 1 && testimony2IsTruthful)
                        || (person2 == 0 && !testimony2IsTruthful);
      TextIO.putln("Person 1 is " + (person1 == 0 ? "a liar" : "truthful")
                   + ". Person 2 is "
                   + (person2 == 0 ? "a liar" : "truthful") );
      TextIO.putln(": Fits assertion 1 and 2 ?" + assertion1 + ","
                   + assertion2);
   }
}
```

6. **Write a program to print out all possible 2 letter words aa to zz:**

Hint use 2 for loops.

10. **MP3 Hints:** "Modulo 26" ; negative integers may bite you.
29 % 26 is 3 (hurrah) … but -3 % 26 is -3 not 23.

11. **What are the values of the array after the following code completes?**

```
// y = row, x = column, assume h = 5
for(int y=0; y < h; y++)
   for(int x = 0; x< h; x++) {
      if( x + y == h)
         A[y][x] = (char)('0' + x%2);
      else
         A[y][x] = ' ';
      A[4-y][0]='?';
   }
return A;
```

11. **Add just one more loop to change all of the outer border cells to be '*'**

12. **Complete the function that returns true iff at least half the entries are positive.**

```
public static boolean positive(double[] data)
{


   for (int i=0; i< data.length; i++) {


   }


}
```

13. **Returns true if there are at least 6 examples where the next array cell is twice the value as the previous one.**

e.g. count({1, 2, 4, 8 , 9, 3, 6, 0, 0, -1, -2 }) will return true.

```
public static boolean count(int[] data) {
   int result = 0;

   for(int i =0; i < _________; i=i+1)
      {

      if( ____________________________)
          result = result +1;



// don't forget the return statement
```

Midterm #1 (Wed):
Name: Rm:
AKa 112 CH
Kf-Ma 120 ARCH
Mb-R 103 TL
S - V 119 MSEB
W-Z 269 EVRT
Conflict(Thurs):
1404 Seibel