CS 125 - Lecture 15

Objectives: Processing 1D arrays; dealing with nulls;

MP3 - due tonight 8pm.

MP4 is out, due two weeks, Monday March 7- challenging and longish;

1. **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];
```

2. **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]);
```

3. 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 _____________;
}
```

4. **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

5. **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 ith movie (in $m)
String[] title = ... //title[i] movie title of ith movie.
```

6. **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++;
   }
}
```

7. **What do the following do? Fix any syntax errors you notice.**

```
new int[6];
new int[6] { 1,2,3,4,5,6 };
int[] a = {1,3,5,7,9,11};
int[] b=null;
b=a;
char[100] myvariable = new char[100];
int len = myvariable.length();
```

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';
}
```

10a. **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;
```

10b. **Add just one more loop to change all of the outer border cells to be '*'**

11. **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++) {


   }


}
```

12. **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
```

7. **What will the following code print?**

```
int[][] data = new int[10][20];
TextIO.putln(data.length);
int[] myrow = data[3];
TextIO.putln(myrow.length);
TextIO.putln(data[3].length);
TextIO.putln(myrow [5]);
myrow[5] is equivalent to data[_____][______]
```

8. **CSI Phone records. (Parallel arrays)**

Print out all entries where a phone call originated from Wisconsir
Some entries may be *null* if the from or to numbers are unknown

```
String from[] = new String[] { "608-123-3311","221-25                ;
String to[] = new String [] { "217-555-6200","217-512
int[] duration = new int[] {1,25,8,23,... };
```

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

11. **What will the following print exactly?**

```
for(int x =3; x<=12; x = x*2) {
   for(int y=x; y>0; y--) TextIO.put("x");
   TextIO.putln();
}
```

9. **Using 2D arrays to represent an image.**

Create a picture of the JVMs memory and use memory pointers to explain why the following code swaps two rows.

```
int[][] pixels;
pixels = new int[480 /*row or 'y' coordinate*/][640 /* column
or 'x'];
//  initialize pixel array : Odd rows are black.
//  Even rows are white
for(int y=0;y< 480; y++)
   for(int x = 0; x< 640; x++)
      if(y % 2 ==0) pixels[___][____] = 0xffffff;
//0xfffff = all white (red=255,green=255,blue=255)
int[] temp = pixels[10];
pixels[10] = pixels[11];
pixels[11] = temp;
```

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.
```