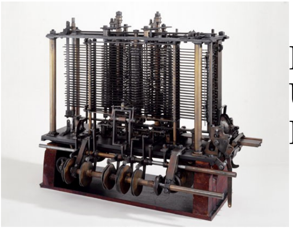

Lab 07 this sheet + 2 pages from G&E.
Introduction to Computer Science
UAlbany, College of Computing and Information
ICSI 201
Fall 2012

INSTRUCTIONS: THERE ARE 4 parts + extra credit.

As usual, make a Lab07 directory and eventually 4 subdirectories: **`Maker, MixerUpper, SmallestFinder, SmallestPutFirster`**

`Maker:`
Make and test a UAlbany style complete Java application that makes a length 5 int array that reproduces the first 5 elements of what Guzdial and Ericson drew in Figure 4.1 It is best to make methods for all this lab's work but that is not mandatory for credit.

Here's "free" code to print an **`int`** array: (You can rename Rosie if you like, for fun.)

```
public static void printIntArray(int[] myParamVar)
{
   int LOCATION = 0;
   while( LOCATION < myParamVar.length)
   {
      System.out.print("In LOCATION " + LOCATION + "   ");
      System.out.print("Rosie has written ");
      System.out.println( myParamVar[LOCATION]);
   }//Whoops..I really forgot something, and was too lazy to
test. You fix my infinite loop. You got what you paid for.
   return;
   //Make the  method return AFTER printing the entire array.
}
```

`MixerUpper`:
The Java expression **`( (int) ( 4.9999 * Math.random() ) )`** makes the computer find a random double precision number between 0.0 and 1.0 and then use it calculate a random integer value among 0, 1, 2, 3 and 4.

Use it in a complete application that, after printing G&E's original array, mixes up its contents and prints it again. (OK if you lose some numbers sometimes).

`SmallestFinder:`
Write a complete application that repeats what MixerUpper does but afterward, finds the smallest int in the array, prints (A) its LOCATION and (B) the actual smallest int in the array.

`SmallestPutFirster:`

Extend your **`SmallestFinder`** (that means add to it without removing anything, except to fix mistakes) so that it also rearranges two array entries so that the smallest int stored in the array is now stored in LOCATION 0 (the first) location, and the int that used to be in LOCATION 0 is now stored in the LOCATION that held the smallest originally.
Ask the TA to explain.
Make it print the array again after putting the smallest first.

`Extra:`

Make a method to sort the array so the smallest int is first, next smallest is second, etc, the largest is last (5th element). Either use ideas from your **`SmallestPutFirster`** work and/or find info about sorting algorithms on the Web or by asking a more advanced CS student or prof.

If you do this, tell or email your TA so he/she will check your work and give you the extra credit (if deserved!).