CSC116
DISCUSSION 11

GINA BAI

LOGISTICS

- Mid semester evaluation survey

- Growth Mindsets - Failure is Part of Success Reflection

  Thursday, June 17th, 11:45pm

- Project 2

  Friday, June 18th, 11:45pm

TOPICS

- Modifying arrays
- Returning arrays
- Array sizes

LAB 10

Write a program called ***IntArray*** that

- Prompt the user for the ***number of values***, n (n > = 1).

  Output "You must enter an integer greater than 0" and reprompt the user if the number of values is not an integer value greater than 0.

- Prompt the user for ***each value and store it in the array***.

  Output "Invalid value" and reprompt if the user enters something other than an integer.

Prompt the user for the ***number of values***, n (n > = 1).

Output "You must enter an integer greater than 0" and reprompt the user if the number of values is ***not an integer value greater than 0***.

```
int numValues = set to some invalid value so we can get into the while loop;
while ( when the number of values is invalid, repeat the following statements ) {
    // Prompt for number of values
    if ( hasNextInt() is false ) {
        // Error message + Reprompt
     }
     else {    // if the input can be read in as an integer
         // read in the input value
          if ( numValues is not greater than 0 ) {
              // Error message + Reprompt
          }
      }
}
// now the numValues is finalized, use it as the array size
```

Prompt the user for each value and store it in the array.

Output "Invalid value" and reprompt if the user enters something ***other than an integer***.

```
int[] values = new int[numValues];
for (int i = 0; i < some limit; i++) {
    // prompt for user input
    while ( hasNextInt() is false ) {
        // Error message + Reprompt
    }
    // read in the input and store it in the array
}
```

LAB11

Write a program called ***SwapArray*** that

- Prompt the user for a string input
- Create an array of characters to store the user input.
- Implement a ***swapPairs*** method to swap the elements at adjacent indexes (for example, elements 0 and 1 are swapped, elements 2 and 3 are swapped, and so on). If the array has an odd length, the final element should be left unmodified.
- Output the array before swap
- Output the array after swap