|  |  |
| --- | --- |
| **Date Assigned: 10/9/15** | **Date Due: 10/13/15** |
| **Unit:** Methodology | **Turn In List:** **1. Terms** |
| *“I will explore and implement the use of arrays in application development.”* | |

**Arrays**

**Content Objectives:** Students will create apps with the use of a powerful and innovative data type.

|  |
| --- |
| **Starter Activity** |
| Create an array of 100 integers and populate it with the numbers 0-100. Print the numbers to the console. Then change the code to fill the array with random numbers between 0-100.  //Array starter  int[] numbers = new int[101];  void settings()  {  fill\_Numbers();  }  void fill\_Numbers()  {  int i = 0;  while(i < 101)  {  int n = int(random(101));  numbers[i] = n;  println(n);  i += 1;  }  } |

|  |  |
| --- | --- |
| **Key Terms:** | |
| Syntax:initialize an array w/ values | int[] numbers = new int[3]; |
| Syntax:initialize an array w/ “new” | numbers[1] = 1; |

|  |
| --- |
| **Assignment:** |
| Complete the following problems with Arrays assuming the following int array. Hint use .length to help achieve results. See the following page for additional information:  **int[] nums = {5,4,2,7,6,8,5,2,8,14};**  **Problem #1:**  // Square each number ((i.e., multiply each by itself)  for (int i 0; i < nums.length; i++) {  nums[i] = nums[i]\*nums[i];  }    **Problem #2:**  // Add a random number between zero and 10 to each number.  for (int i 0; i < nums.length; i++) {  nums[i] += int(random(11));  }  **Problem #3:**  // Add to each number the number that follows in the array. Skip the last value in the array.  for (int i = 0; i < nums.length-1; i++) {  nums[i] += nums[i+1];  }  **Problem #4:**  // Calculate the sum of all the numbers.  int sumOfNums = 0;  for (int i = 0; i < nums.length; i++) {  sumOfNums += nums[i];  }  **Problem #5:**  Write a program that implements a simple rollover. In other words, if the mouse is over a rectangle, the rectangle changes color.  int x = 50;  int y = 50;  int w = 100;  int h = 75;  void setup() {  size(200,200);  }  void draw() {  background(255);  stroke(0);  if (mouseX>x && mouseX<x+w && mouseY>y && mouseY<y+h) {  fill(255, 0, 0);  } else {  fill(255);  }  rect(x,y,w,h);  }  **Problem #6:**  Write a Button class (problem #5 for a non-object-oriented button). The button class should register when a mouse is pressed over the button and change color. Create button objects of different sizes and locations using an array. Before writing the main program, sketch out the Button class. Assume the button is off when it first appears. Here is a code framework:  class Button {  // Button location and size  float x;  float y;  float w;  float h;  // Is the button on or off?  boolean on;  // Constructor initializes all variables  Button(float tempX, float tempY, float tempW, float tempH) {  x = tempX;  y = tempY;  w = tempW;  h = tempH;  on = false; // Button always starts as off  } |

Notes (Points of interest, mistakes, lessons learned, web resources, and thoughts):

|  |
| --- |
|  |