|  |  |
| --- | --- |
| **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.  int[] nums = new int[100];  for (int i=1; i<nums.length; i++) {  nums[i] = int(random(101));  println(nums[i]);  }  int[] nums = new int[101];  for (int i=1; i<nums.length; i++) {  nums[i] = i;  println(nums[i]);  } |

|  |  |
| --- | --- |
| **Key Terms:** | |
| Syntax:initialize an array w/ values | int[] numbers = { 90, 150, 30 }; |
| Syntax:initialize an array w/ “new” | int[] numbers = new int[3];  numbers[0] = 90; // Assign value to first element in the array  numbers[1] = 150; // Assign value to second element in the array  numbers[2] = 30; |

|  |
| --- |
| **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];  println(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] = nums[i] + int(random(0, 11));  println(nums[i]);  }  **Problem #3:**  // Add to each number the number that follows in the array. Skip the last for (int i = 0; i < nums.length; i++) {  nums[i] = nums[i] + nums[i+1];  println(nums[i]);  }  **Problem #4:**  // Calculate the sum of all the numbers.  int sum = 0;  for (int i = 0; i < nums.length; i++) {  sum += nums[i+1];  }  **Problem #5:**  Write a program that implements a simple rollover. In other words, if the mouse is over a rectangle, the rectangle changes color.  boolean button = false;  int x = 50;  int y = 50;  int w = 100;  int h = 100;  void setup() {  size(400,400);  }  void draw() {  background(0,255,0);  if(button){  fill(255,0,0);  text("Button ON: " + str(button),55,40);  fill(0);  rect(x,y,w,h);  } else if (!button) {  fill(255,0,0);  text("Button ON: " + str(button),55,40);  fill(#AA00F5);  rect(x,y,w,h);  }  }  void mousePressed() {  if(mouseX>x && mouseX<x+w && mouseY>y && mouseY<y+h) {  button = !button;  }  }  **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 Name  class Button {  //Class Variables  boolean button;  int x, y, w, h;  //Class Constructor  Button(int tempX, int tempY, int tempW, int tempH) {  x = tempX;  y = tempY;  w = tempW;  h = tempH;  button = false;  }  //Class Attributes  void display() {  if (button) {  fill(255, 0, 0);  text("Button ON: " + str(button), x, y-10);  fill(0);  rect(x, y, w, h);  } else if (!button) {  fill(255, 0, 0);  text("Button ON: " + str(button), x, y-10);  fill(#AA00F5);  rect(x, y, w, h);  }  }  //Class Behavior  void clickButton() {  if (mouseX>x && mouseX<x+w && mouseY>y && mouseY<y+h) {  button = !button;  }  }  } |

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

|  |
| --- |
|  |