|  |  |  |  |
| --- | --- | --- | --- |
| Students are able to: | For example: | Quiz Qs | I need to remember or practice: |
| 1) Declare and initialize arrays and lists | int[] nums = new int[10];  Rectangle boxes = new Rectangle[10];  int[] nums2 = {1,2,3,4}  List<Murloc> murlocs = new ArrayList<Murloc>(); | 1 |  |
| 2) Understand and apply properties of arrays and lists | arrays and lists are objects even if what they hold are primitives. This means normal stack/heap complications apply.  arrays can hold anything, lists only hold objects.  arrays are a fixed size (.length instance field), lists can resize themselves (.size() method). | 2-5 |  |
| 3) Initialize, access, and change individual spots in an array | numbers default to 0, booleans to false, objects to null  nums[4] = 10;  boxes[2] = box1;  int num = nums[2]; | 6-7 |  |
| 4) "Resize" (add or remove elements in) an array | create new array with new size and copy the desired values from the old to the new and storing the reference to the new array | 8 |  |
| 5) Initialize, access, and change individual spots in a list | murlocs.get(0);  murlocs.add(mur1);  murlocs.add(3, mur1);  murlocs.set(2, mur2);  murlocs.remove(3);  murlocs.remove(mur1); | 9 |  |
| 6) Understand and create 2D arrays | 2D arrays are arrays of arrays. Each row in the major array is an entire array.  int[][] chart = new int[10][10]; | 10-11 |  |
| 7) Use loops with arrays and lists | use for loops without going out of bounds  use enhanced for loops  decide which is the correct one to use | 12-13 |  |
| 8) Sort and search an array or list | write & trace:  selection, insertion, & bubble sort  sequential & binary search | 14 - 15 |  |
| 9) Trace complex code involving arrays and lists |  | 16 - 17 |  |
| 10) Write complex code involving arrays and lists |  |  |  |