# Lab Assignment 23 (Sorting)

**NOTE: Your program should be your own work and not a copy of anyone else’s code. Copied code will not earn any points. (Keeping programs unique should not be too hard since you are naming the variables and functions yourself.)**

1. Write a program named **Lab23A** that will fill 3 arrays with random numbers and then use different sort routines to sort each array.
   1. Create a void method to fill an array that will accept an array of integers (only one array) and fill it with random numbers with values between 0 and 500 (inclusive).
   2. Create a void method to perform the selection sort that will accept an array of integers and sort it using the Selection Sort algorithm. (Actually write out the code, don’t just call a premade function, if one exists.) Then print the sorted array all on one line with a label to say which sort was used
   3. Create a void method to perform the bubble sort that will accept an array of integers and sort it with the Bubble Sort algorithm. (Write out the code, don’t just call a premade function.) Then print the sorted array all on one line with a label to say which sort was used
   4. Create a void method to perform the merge sort that will accept an array of integers and sort the array using the Merge Sort algorithm. (Write out the code, don’t just call a premade function.)   
      **Don’t** print the array in this method since it does not work well with the recursion.
   5. In the main method:
      1. Create 3 arrays of 200 integers and call the fill array method 3 times, sending the first array the first time, second array the second time, and third array the third time.
      2. Call the selection sort method sending the first array as the parameter. Record the startTime just before calling the method and endTime just after. Print the time needed for this sort with a label. (Make sure you use **System.nanoTime()**)
      3. Call the bubble sort method sending the second array as the parameter. Record the startTime just before calling the method and endTime just after. Print the time needed for this sort with a label.
      4. Call the merge sort method sending the third array as the parameter. Record the startTime just before calling the method and endTime just after. Print the time needed for this sort with a label.
      5. Print the third array all on one line with a label to say that the merge sort was used. (The other 2 arrays were printed inside their sort methods.)
      6. NOTE: when you print the time for each part, print it with commas. (Either using printf or another formatting method.)
2. Write a program named **Lab23B** that will use the Insertion Sort to sort an array of objects.
   1. Create a secondary class named **Student** with the following:
      1. 3 instance variables to hold the student first name (String), last name (String), and grade point average (double)
      2. A constructor that receives 3 parameters and fills in the instance variables
      3. A double method (no parameters) that returns the grade point average
      4. A String method named **toString** (no parameters) that returns a String containing the three instance variables with labels.
   2. Back in the main class:
      1. Write a void method that receives an array of Student objects as a parameter and uses the Insertion Sort algorithm to sort the array by the Student GPAs.   
         (Your method should have all the steps of the Insertion Sort, not a shortcut.)
      2. In the main method:
         1. Declare an array of 12 Student objects
         2. Read the data for Student object from the text file (Lab23B.txt), create the object, and add it to the array
         3. Call the void sorting method sending the array as a parameter
         4. Print the array
3. Write a program named **Lab23C** that will use the Bubble Sort to sort an array of objects.
   1. Create a secondary class named **Software** with the following:
      1. 2 instance variables
         1. title (String)
         2. price (double)
         3. number of players (int)
      2. A constructor that receives a String, a double and an int as parameters and fills in the instance variables
      3. A double method (no parameters) that returns the price
      4. An int method (no parameters) that returns the number of players
      5. A String method named **toString** (no parameters) that returns a String containing the instance variables with labels.
   2. Back in the main class:
      1. Write a void method that receives an array of Software objects as a parameter and uses the Bubble Sort algorithm to sort the array by the prices.   
         (Your method should have all the steps of the Bubble Sort, not a shortcut.)
      2. Write a second void method that receives the array and (bubble) sorts them by number of players
      3. In the main method:
         1. Declare an array of 5 Software objects
         2. Read the data for each Software object from the text file (Lab23C.txt) then create the object and add it to the array
         3. Call the first sorting method sending the array as a parameter
         4. Print the array
         5. Call the second sorting method sending the array as a parameter
         6. Print the array