Final Exam Practice Sheet

Define “class”. Define “object”.

What Java code is required for a class to properly implement the Comparable interface?

Evaluate the following Boolean expressions **(True or False)**

```
a. 5 % 9 == 5 or (6 * 3 / 4 > 4  and 20 / 6 != 3)

   Answer:  True

b. 5 % 9 == 6 and 5 * 4 / 3 > 4 or 6 / 20 <= 3


c. 4 < 3 and 2 == 2 and 3 > 9 and 34532 / 324 > 293


d. 4 <= 3 and (2 == 2 or 3 / 6 >= 9) or 3 > 0
```

Evaluate each of the following expressions. Values without a decimal point are integers, values with a decimal point are real numbers. Show the value of each expression to the right of the question – **PLEASE CIRCLE YOUR ANSWER**. Also, show your work where possible for partial credit.

```
   a. 5 / 3 * 2.0 / (4 % 3) + 5  (answer:  7.0)


   b. 6.0 * 7 * 2 / 4 + 5 * 11
   c. 6 % 4 + 9.0 / 4 + 2 * 5
```

```
   d. 22 / 44 + (3 % 31)
```

What is the output of the code segment below (show answer in space to right)?

```
double a = 0;

while (a <= 10)
{
   System.out.print (a + “ “);
   a--;
}
```

Write a method which is passed three Strings. The method returns a 3 if the strings are in alphabetic ascending order **(apple, boat, car**), a 2 if the strings are in alphabetic descending order, and a 1 if the strings are in no particular order. You may assume the values of the three strings will all be different☺**.**

Write a method called **createArray** that is passed a single integer value (guaranteed to be greater than 0). The method should create an array of type **int**, using the parameter value for the length. Fill it with values starting at 1 (so element 0 will have a value of 1, element 1 will have 2, etc.), then return the array.

Yep. You guessed it. Write the 6 standard methods every self-respecting class should have for the Song class. Here’s the driver class:

```
public class SongTester
{

      public static void main(String args[])
      {
            private Song[] songs;

            songs = new Song[3];

            songs[0] = new Song(); //defaults to: “Untitled” for name of song, and
                                    // “Undetermined” for name of artist…
            songs[1]  = new Song(“I Can’t Stop Loving You”, “Ray Charles”);
            songs[2] = new Song(“Daylight”, “Matt & Kim”);

            System.out.println(songs[0]); //toString called via this statement

            SortSearchUtil.insertionSort(songs);  // Make the Song object Comparable

            System.out.println(“After sorting”);
            System.out.println(songs[0]); // Is the first element the right value?

            if (SortSearchUtil.linearSearch(songs, new Song(“Chain Gang”, “Sam Cooke”))
            {
                  System.out.println(“Already on file.”);
            }
            else
            {
                  System.out.println(“Not on file.”);
            }



      }// end method


}//end class SongTester
```

Which is more efficient for an array of elements:

insertionSort or selectionSort..? Why?

linearSearch or binarySearch…? Why?

What condition must be met for binarySearch to work?

What is the time-complexity formula for binarySearch?

Write a standard array traversal structure using a ‘for’ loop:

Write a standard array traversal structure using a ‘for-each’ loop:

Write a ‘displayMenu()’ method that presents 4 menu options:

option 1 – Get the radius of a circle
option 2 -- Get the area of a circle
option 3 -- Get the circumference of a circle
option 4 -- Quit

and returns a value of type int. Use a do-while loop and insure the user’s input is valid:

```
public static void main(String[] args)
{
      // Create a 'switch' structure for menu
      // option 1 -- call calcRadius method
      // option 2 -- call calcArea method
      // option 3 -- call calcCircumference method
      //  (all methods accept a Scanner and return a double –
       //   display the double after it's returned)

          int option;
          Scanner kb = new Scanner(System.in);
          option = displayMenu(kb);

          while (option != 4)
          {
             //  Your code goes here:













             option = displayMenu(kb);
          }
       }
```