Worksheet W0: Object-Oriented Programming (10 points)

Out: 2025 January 14 (Tuesday evening)

Due: 2025 January 16 (Thursday end of day [2359 CDT according to D2L]). ***No late submissions will be accepted***

## What to submit?

Upload an MS Word or PDF file. If you take photos you could zip them, but I’d prefer that you paste them into Word/PDF docs. Upload exactly one file to the designated D2L folder. Some questions may be answered directly on the word file. Other questions can be answered on a piece of paper, and you take a picture of your paper and insert the picture in the word file. The worksheet is available online and is open now.

# Exercise 1: Arrays of Primitives (0.5 pt each)

1. Declare a reference variable for an array of doubles.  Name the array prices.

double[] prices;

1. Instantiate the prices array to hold 50 doubles.

prices = new double[50];

1. Set the array element of prices that is at index three to 13.99.[[1]](#footnote-1)

prices[2] = 13.99;

1. Write a for loop to count the number of values in the prices array that are greater than 10.

double sum = 0;

for (int i = 0; i < prices.length; i++) {

if (prices[i] > 10) {

sum += prices[i];

}

}

System.out.println(sum);

# Exercise 2: Comparing Objects (3 pts)

Write an equals method for the Car class given here.  Two Cars are equal if their Vehicle Identification Numbers (VIN) are the same.  HINT:  The String class has an equals method that can be called from the Car class equals method.

public String getVin() {

            return vin;

        }

        public String getColor() {

            return color;

        }

        public Boolean isEqual(Car obj) {

            return obj.getVin().equals(this.vin);

        }

Text

Description automatically generated

# Exercise 3: UML diagrams (3 pts)

Draw the UML diagram for the Car class from Exercise 2 after including the equals method.

Chart completed from Lucid.app

A screenshot of a car

Description automatically generated

# Exercise 4: Arrays of Objects (0.4 pts each)

1. Declare and instantiate an array of 40 cars using the Car class.  Name the array dealerCars.

Car[] dealerCars = new Car[40];

1. Write Java statements to add three cars to the dealerCars array.

dealerCars[0] = new Car("123", "black");

dealerCars[1] = new Car("456", "blue");

dealerCars[2] = new Car("789", "red");

1. Make a hole at the beginning of the dealerCars array by moving the cars in the array up by one position.

for (int i = dealerCars.length - 1; i > 0; i--) {

dealerCars[i] = dealerCars[i - 1];

}

1. Add another car to the beginning of the array.

dealerCars[0] = new Car("988","pink");

1. Write a for loop to display all the cars in the dealerCars array. Add any necessary methods to the Car class.

for (Car car : dealerCars) {

if (car != null) {

System.out.println("VIN: " + car.getVin()+ " Color: "+ car.getColor());

}

}

1. Remember that array indices start at 0. Because some people would call the element prices[0] the “first” element of the array, others (incl. me) would call the “zeroth” element, some people would call this the “fourth” element of the array, others would call it the “third” element. [↑](#footnote-ref-1)