Week4Lab – 10 pts

Pre-lab questions

1. The best way to get comfortable with loops is to write them! Write just the loops (and any required, dependent variables) to do the following:
   a. Count down from 10 to Blast Off!, printing each number on its own line.

```
int second = 10;
while(second >= 0){
    if (second == 0){
        System.out.println("Blast off");
    }
    else {
        System.out.println(second);

    }
    second--;
}
```

   b. Replace the lowercase letter e from each user entered String, replacing it with an underscore and printing the new String. Allow the user to enter as many Strings as they want. Choose a good sentinel value.

```
boolean done = false;
while (! done){

System.out.print("Enter any word and enter 0 when finished: ");
                String str = scan.nextLine();

                if (str.equals("0")){
                    done = true;
                }
                else{

                  System.out.println(str.replace('e','_'));
                }
            }
```

   Count up by a given number from 0 to another given number. For example the user may enter 5 and 30 and the loop should produce: 5, 10, 15, 20, 25, 30 then stop.

```
Scanner scan = new Scanner(System.in);
System.out.print("Enter the number: ");
int num = scan.nextInt();
for (int i = num; i <= 30; i += 5){
```

```
        System.out.println(i);

    }

}
```

c. Rewrite the while loop in a. to a for loop.

```
for (int second = 10; second >= 0; second--){

        if (second == 0) {
              System.out.println("Blast Off");

        }
        else {
              System.out.println(second);

        }
}
```

d. Write a for loop that adds the even numbers from 0 to 100, then prints the final result only.

```
  int sum = 0;
for (int num = 0; num <= 100; num += 2){

         sum += num;
}

  System.out.println(sum);
```

Choose one of the following do develop into a program that uses at least one class outside of the driver class that contains the main method. Your solution MUST use an ArrayList of objects created from your custom class. Once chosen, do the following:

Understand the problem (restate in your own words, make any assumptions clear):

UML diagrams of any classes needed, including the one with main:

Pseudocode of each class:

Name of files (.java) submitted:

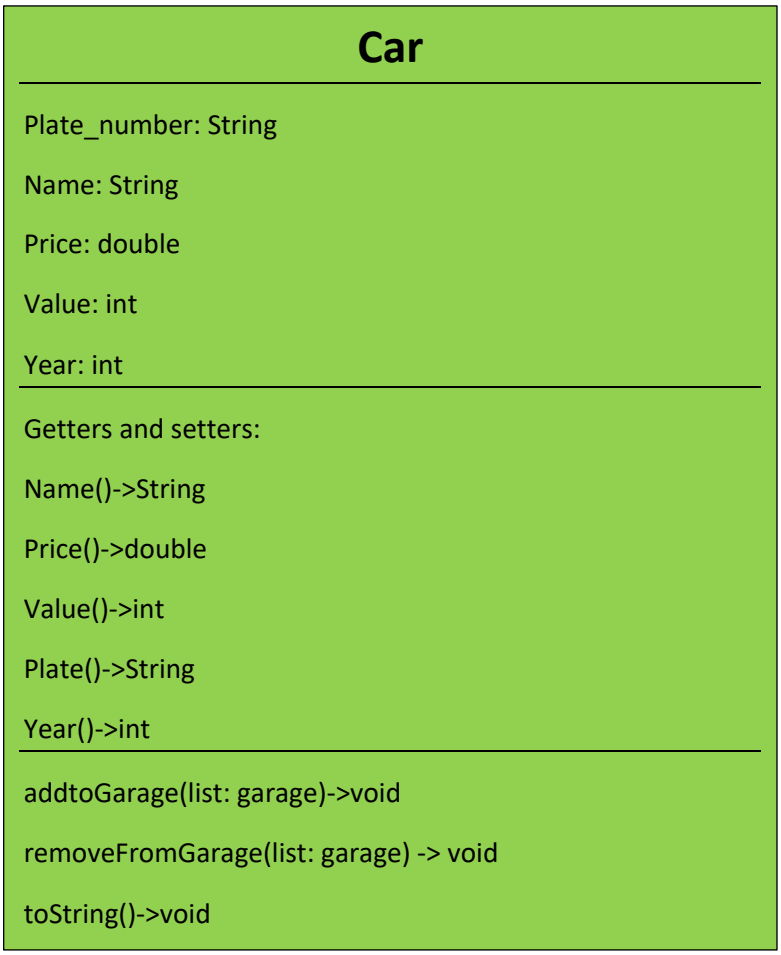
Screenshots of at least three separate runs of your program with test data:

Garage and Cars

Write a Car class that includes key ways we describe cars. It has to include either price or value (it can include both). Be sure to determine the getters and setters that need to be written, and write the toString() method appropriately so when you 'print' a car, it makes sense.

The driver class will contain a garage (ArrayList) that contains several cars. The user must be able to add a car to the garage, print the contents of the garage, get the value of the cars in the garage, remove a car from the garage (like selling it).

We need a car class that has a price a value and other attributes describing cars. The class should behave in such a way so that a car can have either a price, a value, or both. It should have a method to add the car to a garage (an Arraylist) and another method to remove one. All attributes will have a getter and a setter.



Car.java

CarGarage.java

Car Class Pseudocode:

```
class Car:

def Car(String plate, String name, double price, int value ,int year){
    this.plate = plate;
```

```
        this.name = name;
        this.price = price;
        this.value = value;
        this.year = year;
    }



// Getters and setters will continue for all attributes
def getName() {
        return name;
    }
…

 def  setYear(int year) {
        this.year = year;
    }

addtoGarage(garage) {
        printf("%s added to garage\n", this);
        garage.add(this);
    }


def  removefromGarage(garage) {

        System.out.printf("%s removed from garage\n", this);
        garage.remove(this);
    }
```

```
    def toString(){

        String info = f"Car(Plate: %s, Name: %s, Year: %d, Price: %f, Value: %d)",
                                    this.plate, this.name, this.year, this.price, this.value);
        return info;

    }

}
```

```
Enter the price of the first car: 23984.34
Enter the value of the first car: 34000
Do you wish to add this car to the garage? Y/N? y
Car(Plate: null, Name: null, Year: 0, Price: 23984.340000, Value: 34000) added to garage

Enter the intial price of the second car: 128343.34
Do you wish to add this car to the garage? Y/N? y
Car(Plate: null, Name: null, Year: 0, Price: 128343.340000, Value: 0) added to garage

Enter the intial value of the third car: 72634
Do you wish to add this car to the garage? Y/N? n

Let's add some more cars. We added one with defaults for you to modify
Enter the plate number of the 4th car: wrt-1023
Enter the name: HONDA CRV
Enter the year: 2019
Enter the price: 38912.86
Enter the value: 40000

Car 4's name is HONDA CRV
Car 4's license plate is WRT-1023
Car 4's price is 38912.860000
Car 4's year is 2019
Car 4's value is 40000
Car(Plate: WRT-1023, Name: HONDA CRV, Year: 2019, Price: 38912.860000, Value: 40000) added to garage

Cars have been added. The number of cars in the garage is 3.

10 minutes later a customer came in to buy car 4.
Car(Plate: WRT-1023, Name: HONDA CRV, Year: 2019, Price: 38912.860000, Value: 40000) removed from garage

The number of cars in the garage is now 2
```

```
Enter the price of the first car: 23745.12
Enter the value of the first car: 28000
Do you wish to add this car to the garage? Y/N? n

Enter the intial price of the second car: 10234.34
Do you wish to add this car to the garage? Y/N? y
Car(Plate: null, Name: null, Year: 0, Price: 10234.340000, Value: 0) added to garage

Enter the intial value of the third car: 20394
Do you wish to add this car to the garage? Y/N? y

Car(Plate: null, Name: null, Year: 0, Price: 0.000000, Value: 20394) added to garage

Let's add some more cars. We added one with defaults for you to modify
Enter the plate number of the 4th car: ASD-5218
Enter the name: Toyota Camry
Enter the year: 2016
Enter the price: 37453.23
Enter the value: 41000

Car 4's name is TOYOTA CAMRY
Car 4's license plate is ASD-5218
Car 4's price is 37453.230000
Car 4's year is 2016
Car 4's value is 41000
Car(Plate: ASD-5218, Name: TOYOTA CAMRY, Year: 2016, Price: 37453.230000, Value: 41000) added to garage

Cars have been added. The number of cars in the garage is 3.

10 minutes later a customer came in to buy car 4.
Car(Plate: ASD-5218, Name: TOYOTA CAMRY, Year: 2016, Price: 37453.230000, Value: 41000) removed from garage

The number of cars in the garage is now 2
```

```
Enter the price of the first car: 92384.45
Enter the value of the first car: 102394
Do you wish to add this car to the garage? Y/N? y
Car(Plate: null, Name: null, Year: 0, Price: 92384.450000, Value: 102394) added to garage

Enter the intial price of the second car: 304982.34
Do you wish to add this car to the garage? Y/N? n

Enter the intial value of the third car: 409384
Do you wish to add this car to the garage? Y/N? n

Let's add some more cars. We added one with defaults for you to modify
Enter the plate number of the 4th car: dfl-1234
Enter the name: Lamborghini
Enter the year: 2022
Enter the price: 305123.34
Enter the value: 405123

Car 4's name is LAMBORGHINI
Car 4's license plate is DFL-1234
Car 4's price is 305123.340000
Car 4's year is 2022
Car 4's value is 405123
Car(Plate: DFL-1234, Name: LAMBORGHINI, Year: 2022, Price: 305123.340000, Value: 405123) added to garage

Cars have been added. The number of cars in the garage is 2.

10 minutes later a customer came in to buy car 4.
Car(Plate: DFL-1234, Name: LAMBORGHINI, Year: 2022, Price: 305123.340000, Value: 405123) removed from garage

The number of cars in the garage is now 1
```

Airplane Seats

Write a Seat class that contains information on a given seat on an airplane. The seat is available or sold. If it is sold, it knows the confirmation number. It always knows its value (or price) and its type. You get to

name the types, but make sure that “desk” is one of them! Determine the getters and setters and any other methods you need, but be sure to write toString() because you will be printing out the seats.

The driver class will contain an ArrayList of all the seats. Use a loop or loops to generate at least 100 seats in total. The user must be able to see all the seats that are available, and all that are available of a certain type (like “desk”), choose one getting a confirmation number, then be able to view all seats with that confirmation number.

Cooking and Calories

Write an ingredient class so that a given ingredient knows its unit and calories, such as 100 calories and 1 cup, as well as its name and type, such as protein, carb or fat.

The driver class for this will allow the user to create ingredients from a recipe, see them as a list, one ingredient per line, formatted as expected for a recipe. They will also be able to get the total calories for the whole recipe and be able to enter the number of servings and get the calories per serving. The user should also be able to get a list of ingredients by type, ex. printing only the protein ingredients.