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:
   1. Count down from 10 to Blast Off!, printing each number on its own line.

**int** i = 11;

**while** (i >= 1) {

i--;

**if** (i == 0) {

System.***out***.println("Blast Off!");

} **else** {

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

}

}

* 1. 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.

**while** (**true**) {

System.***out***.println("Enter the string");

String word = keyboard.nextLine();

**if** (!word.contains("e")) {

**break**;

} **else** {

String newWord = word.replace("e", "\_");

System.***out***.println(newWord);

}

}

* 1. 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 keyboard = **new** Scanner(System.***in***);

System.***out***.println("Enter first number");

**int** firstnumber = keyboard.nextInt();

System.***out***.println("Enter last number");

**int** lastnumber = keyboard.nextInt();

**for** (**int** i = firstnumber; i <= lastnumber; i = i + firstnumber) {

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

}

* 1. Rewrite the while loop in a. to a for loop.

**for** (**int** i = 10; i <= 10; i--) {

**if** (i < 0) {

**break**;

}

**if** (i == 0) {

System.***out***.println("Blast Off!");

} **else** {

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

}

}

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

**int** evenSum = 0;

**for** (**int** i = 0; i <= 100; i++) {

**if** (i % 2 == 0) {

evenSum = evenSum + i;

}

}

System.***out***.println(evenSum);

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:

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.

Understand the problem (restate in your own words, make any assumptions clear):   
I understand the problem I have to make two classes ingredients and driver. In ingredients class I have to set the name, unit and calories and in constructor I have to define them so whenever it call the value set to that parameter. Then I use getters to get the name unit and calories from user and tostring to print all the things together. Driver class will create the list of ingredients class which will store the values of units names and calories. Once we enter the servings it will print out the calories for everyone.

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

Graphical user interface, text

Description automatically generated

Pseudocode of each class:

Ingredients class

In ingredients class I have to set the name, unit and calories and in constructor I have to define them so whenever it call the value set to that parameter. Then I use getters to get the name unit and calories from user and tostring to print all the things together.

Driver Class

It will create a list of ingredients it will ask user to print out the name units and calories. Then after entering these data once it enter done it will print all the data. Then it ask for the ingredients type if you enter protein it will print recipe with proteins list data.

Name of files (.java) submitted: Ingredients.java Driver.java  
Driver.java Ingredients.java

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

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

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).



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.