LSUS | Department of Computer Science
318.795.4266 (Fax) 318.795.2419
One University Place
Shreveport, LA 71115-2399

CSC 135, Spring 2020
Dr. Urska Cvek
Lab 5
Assigned: 3-31-20
Due: 4-25-20 by 7 am (Saturday) – THIS DEADLINE IS EXTENDED BY 1 WEEK

Problem 5.1
Write a program that prompts the user to input a "word" representing a binary number (a sequence of 0s and 1s for example, the user would enter 01022). First, your program should verify that it is indeed a binary number that is only when the number contains 0s and 1s. If that is not the case, your program should print a message that the number is not a valid binary number and keep prompting the user for a new "word" until a "word" representing a valid binary number is entered by the user.

After the user enters the valid word, the program should count how many 1s are in that word and output the count and print that out.

Problem 5.2
Write a JavaFX application that allows the user to pick a set of pizza toppings using a set of check boxes. A plain pizza (dough, sauce, cheese included) costs $10 (your initial prize for the pizza). Each additional topping costs 50 cents. The user is able to choose one or more toppings to be added to the pizza (by clicking the check boxes) and this would then display the accurate cost of the pizza with those toppings included.

Topping choices are all yours, but you have to include at least 6 different ones (corresponding to 6 check boxes).

Problem 5.3
Using the *Die* and *RollingDice* classes from Chapter 4, design and implement a class called *DieTriplet* that represents three dice as an object. Review the methods of the *Die* class, which represents a single die object, which can be rolled, or its value changed, or looked-up (using the *setFaceValue* or *getFaceValue* methods), together with printing of the object using the toString method. For the *DieTriplet* class include methods to

- set the individual die face values (one setter method and passing in three parameters, one for each die)
- three getter methods that return the individual die face values (one get method for each of the dice)
- roll the dice (one roll method that rolls each of the dice inside of the body of the roll method)
- return the sum of the three dice (one method that returns an integer that represents the sum of the three face values)

Your driver class can be named *RollTriplet* and it should instantiate one *DieTriplet* object. It should then roll this triplet 5,000 times and count the number of three ones that occur in each of the 5,000 repetitions (each die would have a one as the face value). At the end of the game display the number of three ones that occurred over the 5,000 rolls.

You are required to use some kind of loop to achieve this, at least one (if not more).

In this assignment you are free to choose the names of your classes – they have to follow the Lab Submission Guidelines, though.

Answer above question as programming projects in NetBeans.
Remember to use the Lab Submission Guidelines when submitting your work!