

Can you Remember the Lab 1?

In Lab 1, we got three problems to deal with. In this assignment, you are going to add NUnit testing to these problems.

Problem 1:

Design a BankCharges Class which will perform the following operations:

A bank charges \$10 per month, plus the following check fees for a commercial checking account:

- \$.10 for each check if less than 20 checks were written
- \$.08 for each check if 20 through 39 checks were written
- \$.06 for each check if 40 through 59 checks were written
- \$.10 for each check if 60 or more checks were written

The bank also charges an extra \$15 if the account balance falls below \$400 (before any check fees are applied). Design a class that stores the ending balance of an account and the number of checks written. It should also have a method that returns the bank's service fees for the month.

Use proper GUI controls to take input about the aforementioned parameters and use proper controls to view the results as well.

To test the BankCharges class and the method which is calculating these charges, use demo input and out and use NUnit testing methodologies to check the accuracy of the BankCharges class and the methods.

Problem 2: (10 Marks) (1%)

The Fast Freight Shipping (FFS) Company charges the following rates:

Weight of Package (in KG)	Rate per 500 Miles Shipped
2 kg or less	\$1.10
Over 2kg but not more than 6 kg	\$2.20
Over 6 kg but not more than 10kg	\$3.70
Over 10 kg	\$4.80

The shipping charges per 500 miles are not prorated. i.e. if a 2 kg package is shipped 550 miles, the charges would be \$2.20.

Design a class with GUI controls that stores the weight of a package, and has a method that returns the shipping charges.

To test the class and the method which is calculating these charges, use demo input and out and use NUnit testing methodologies to check the accuracy of your code.

Problem 3:

Write a class that will predict the size of a population of organisms. The class should store the starting number of the organisms, their average daily population increase (as a percentage) and the number of days they will multiply. The class should have a method that uses a loop to display the size of the population for each day.

Input validation: Do not accept a number less than 2 for the starting size of the population. Do not accept a negative number for the average daily population increase. Do not accept a number less than 1 for the number of days they will multiply. Generate Message for the invalid input and show that to the users.

Use GUI to interact with the class and population of the organisms.

To test the class and the method which is calculating the population of the organisms, use demo input and out and use NUnit testing methodologies to check the accuracy of your code.

Deliverables:

Like other two assignments, this assignment 3 is a team work as well. Each team member should solve only one problem. E.g. If there are three members in a team, you have to randomly select one problem, use your lab 1 submission and then add NUnit testing and its methodologies to perform the unit testing. If there is two members in a team, randomly select two out of the three problems, select one for each member and follow the process I stated in my last sentence.

For submission, you have to push your solution of the problem with the NUnit testing code in your team Github repository, but push it in your branch, keep it over there. Each member should push and keep the code for this assignment in their own branch. Finally, create a word file, add each member's name in it and their related branch where the member's put their code and upload it in the LEA.

Submission Deadline:

November 12, 2023 by noon.