# Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Tech Check# 1

## A Simple Test Driven C# Project

Use the two zipped folders which are uploaded on Tech Check 1 Drop Box. One of them contains the unit tests code, the second contains the project that you are going to work on it:

1. Create a word document file, save it as your first name and inside it, copy a captured image which shows that there is a reference added between the Unit Test project and the Class Library

**(1 point \_\_\_\_\_)**

2. Get both Unit tests ( AnnualSalaryTest() & HourlyWageTest() ) passing. And copy captured images inside the word document file showing the test explorer results. **(1 point \_\_\_\_\_)**

3. In your word document file, explain why we need to create a reference between the two projects files. **(2 points \_\_\_\_\_)**

4. From the unit test codes file, copy the first two test methods for negative and zero values in the GetHourlyWage() and GetAnnualSalary methods. Note that you expect an exception to be thrown and your Assert is on the error message received. Make sure that the two test methods pass. Save a captured image for the passed test results in your word document file. **(3 points \_\_\_\_\_)**

5. From the unit test codes file, copy the remaining two test methods and let them pass the test by applying the following:

**Add a function that calculates the total amount of tax withheld from an employee’s weekly salary.**

**The net tax amount is calculated by computing the tax amount, minus a per-dependent deduction from the total tax withheld. The user will enter their pre-tax weekly salary amount and the number of dependents they wish to claim. The program will calculate and output the amount of tax, the dependent tax deduction, net tax amount, and the user’s final take-home amount.**

**Tax amount is calculated at 25.0% of the weekly salary. The tax deduction for dependents is calculated at 5.0% of the employee’s salary per dependent**.

Your **EmpAccount** function will take two parameters: **weeklySalary** [double] and **numDependents** [int]. The function will return a POCO called **empData** with four double parameters, one for each of the required values. For example, calling **EmpAccount(1500, 3)** should return an **EmpData** with the following properties and values:

**EmpData** {

TaxAmount: 375.0

DependentDeduction: 225.0

NetTaxAmount: 150.0

TotalTakeHome: 1350.0

}

**Note:** **EmpData** should be in its own .cs file in the Calculator project.

**(1 pt – EmpData Class in own file in Calculator Project NOT Test Project) \_\_\_\_\_**

**(4 pts – 2 points for each provided Unit Test passing) \_\_\_\_\_**

5. Include your name and W number as a comment line inside the EmpData.cs file. **(1 point)**

**6. Upload the word document file and the Solution project on Brightspace.**

# Final Score (Out of 13) **\_\_\_\_\_\_\_**