Fundamental of Programming in C#

Day 6 Quiz

1. You are asked to implement income tax calculator for Singapore. The tax rate table is given in Figure 1. (For YA 2014 to YA 2016)

No	Chargeable Income	Income Tax Rate (%)	Gross Tax Payable ($)
0	First $20,000	0	0
	Next $10,000	2	200
1	First $30,000	-	200
	Next $10,000	3.5	350
2	First $40,000	-	550
	Next $40,000	7	2,800
3	First $80,000	-	3,350
	Next $40,000	11.5	4,600
4	First $120,000	-	7,950
	Next $ 40,000	15	6,000
5	First $160,000	-	13,950
	Next $ 40,000	17	6,800
6	First $200,000	-	20,750
	Next $120,000	18	21,600
7	First $320,000	-	42,350
	Above $320,000	20	

Figure 1. Singapore Resident Tax Rate (source: IRAS)

As an illustration, someone with $100,000 annual income will fall under the bracket no 3 and have to pay $3,350 for the first $80,000 of the taxable income and will be taxed at 11.5% for the remainder of the taxable income. The calculation would be 11.5% * ($100,000 – $80,000) + $3,350 = $2,300 + $3,350 = $5,650.

You are asked to write the program in modular fashion by implementing the method as prescribed in the specification below.

Specification of the methods

Method Name	***Description***
Main (code given)	The main method should: - Get the taxable income from the user - Get the tax bracket of the user - Calculate the taxable income based on the tax bracket - Print the result
AskForIncome	This method takes no argument and return an **integer** that contains the taxable annual income of the user. This method should: - Prompt the user for annual taxable income. "Please enter your annual taxable income: " - Return the entered income as an integer
GetTaxBracket	This method takes one argument: the annual income of the user and returns an integer that indicates the tax bracket index that the user belong to. The logic for this method should be: - Look for the **largest** index in the minIncome array where the minIncome is **smaller** than the annual income given - Return the largest index found - If the income is less than 20,000, no index would be found, return -1.

CalculateIncomeTax	This method takes two arguments:
	- The annual income
	- The tax bracket index
	And return the payable tax (use double data type for the payable tax)
	
	The logic for this method should be:
	- If the tax bracket index is -1, then no tax is payable.
	- Take the following values from the different arrays:
	  ○ Minimum income (from minIncomeArray)
	  ○ Tax rate (from taxRateArray)
	  ○ Base Payable Amount (from basePayableAmountArray)
	- Calculate the payable tax by using the formula:
	
	Payable tax = (annual income – minimum income) * tax Rate + base payable amount
	- Return the payable tax amount

PrintResult	The arguments for this method are: - The taxable annual income

	- The payable tax amount The console output of this method should match the sample output given. The income and tax amount should be formatted in currency format.

The Main method for the program is given as displayed below:

```
using System;

namespace TaxCalculator
{
    class Program
    {
        //these arrays is visible in all the static method,
        //so you can use them in your method implementation

        static int[] minIncomeArray = new int[]
            { 20000, 30000, 40000, 80000,
              120000, 160000, 200000, 320000 };
        static double[] taxRateArray = new double[]
            { 0.02, 0.035, 0.07, 0.115,
              0.15, 0.17, 0.18, 0.20 };
        static int[] basePayableAmountArray = new int[]
            { 0, 200, 550, 3350,
              7950, 13950, 20750, 42350 };

        static void Main(string[] args)
        {
            int annualIncome = AskForIncome();
            int taxBracket = GetTaxBracket(annualIncome);
            double taxPayable =
                CalculateIncomeTax(annualIncome, taxBracket);
            PrintResult(annualIncome, taxPayable);
        }

        //YOUR CODE HERE
```

```
}
```

Sample outputs from multiple executions:

```
Please enter your annual taxable income: 100000
For taxable annual income of $100,000.00, the tax payable amount is $5,650.00


Please enter your annual taxable income: 0
For taxable annual income of $0.00, the tax payable amount is $0.00

Please enter your annual taxable income: 1000000
For taxable annual income of $1,000,000.00, the tax payable amount is $178,350.00
```

2. The Caesar cipher is one of the earliest known ciphers in the world.

The encryption method used in Caesar cipher is described as follows. It involves shifting each letter in a plain text by three letters forward:

A -> D, B -> E, C ->F, etc…

At the end of the alphabet, it wraps around the end, therefore:

X-> A, Y -> B, Z ->C.

Only letters are encrypted, numbers and other characters remain the same during encryption.

For example, this string “10 ZEBRAS” will be encrypted as “10 CHEUDV”.

You are required to write a simple Caesar cipher program, which can read input sentence from console, encrypt and decrypt it, as well as print the output to console.

Method Specification

The program is written as a class with a static Main() method shown in the following figure which uses other methods that has to be written.

Method Name	**Specification**
ReturnUpperInputSentence	• This method accepts no argument and returns a string • This method should write a string on the console “Please enter the sentence: ” and then capture the user input • The raw input should be converted to upper case and return it as the return value of the method
EncryptSentence	• This method accepts a string argument and return an encrypted string • This method involves shifting each letter in a plain text by **three** letters forward: A -> D, B -> E, C ->F, etc… At the end of the alphabet, it wraps around the end, therefore:

	X-> A, Y -> B, Z ->C. • Only letters are encrypted, numbers and other characters remain the same during encryption. • You can use char.IsLetter() to check if a character is letter or not. • This method returns the encrypted sentence.
PrintEncryptedSentence	• This method accepts one argument that is the encrypted sentence and returns no value. • This method should write a string on the console “The encrypted sentence is : ” and prints the encrypted sentence on the console.
DecryptSentence	• This method accepts one argument that is the encrypted sentence and returns the decrypted sentence.
PrintDecryptedSentence	• This method accepts one argument that is the decrypted sentence and returns no value. • This method should write a string on the console “The decrypted sentence is : ” and prints the decrypted sentence on the console.

Source code of Main method:

```
class Program
{
    static void Main(string[] args)
    {
        string upperPlainText = ReturnUpperInputSentence();

        string encryptedText = EncryptSentence(upperPlainText);

        PrintEncryptedSentence(encryptedText);

        string decryptedText = DecrpytSentence(encryptedText);
```

```
            PrintDecryptedSentence(decryptedText);

            Console.WriteLine("\nType any key to exit.");
            Console.ReadLine();
        }
        //YOUR CODE HERE

    }
```

Code snippet which contain the Main() method

You are required to complete all the FIVE methods.