ASSIGNMENT 1 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	Unit 1: Programming		
Submission date		**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Luu Tran Quang Vu	**Student ID**	BH01337
Class	SE07101	**Assessor name**	Nguyen Thanh Trieu

Student declaration

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

Student's signature	VU

Grading grid

P1	P2	M1	D1

☼ Summative Feedback:		☼ Resubmission Feedback:
Grade:	Assessor Signature:	Date:

Lecturer Signature:

TABLE CONTENTS

I. INTRODUCTION

- I currently work for ABC Software, an independent software development company that designs and builds bespoke software solutions for companies of various sizes, operating in a variety of industries. The software they design uses a variety of technologies, from simple standalone programs to large web-based applications. ABC Software asked for a simple program to calculate monthly water bills. The company's CEO reviewed the client's requirements and determined that this was the right project for me to take on. The company wants to see how I use and apply development environments and code standards.
- The customer had not specified any Graphical User Interface, but the CEO had left the design of the program entirely up to me.
- After the program was built, the CEO asked me to report back to them on how I designed and developed the necessary algorithms and how I converted these algorithms into the final program and Point out any problems you encounter.
- As part of my report, the CEO wanted me to create a presentation for the development team showing the different programming paradigms available as well as the debugging tools I used along the way. develop the program so the development team can review my progress.

II. BODY

A. Basic algorithms to perform an operation and outline the process of programming an application

1. Basic algorithms:

- Linear Search
- Binary Search
- Quick Sort
- Selection Sort

2. Process of programming an application:

- Determine requirements
- Design
- Program
- Testing and Debugging
- Deploy and maintain

B. Algorithm definition:

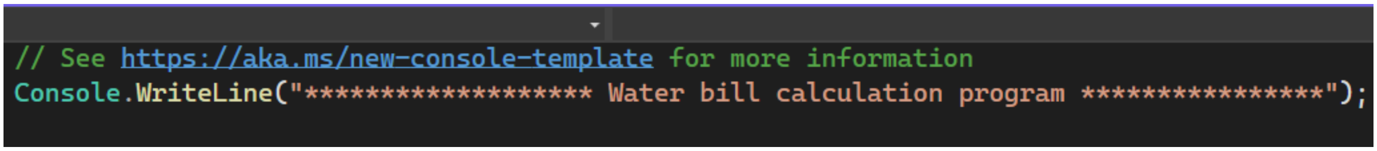
- An algorithm is a method to solve a problem. To put it more simply, each math problem is like a treasure chest (results, answers), and the key to open that chest is the "algorithm". If you use the wrong key, you can still open the chest, but it will take a long time, or if you open the chest, the treasure inside will be distorted and incomplete. Using the right key will help you get the treasure easily and quickly. Of course, each chest will always need a different type of key, just like a math problem always has certain algorithms. There is no key that can open all the boxes, just as there is no algorithm that can solve all the problems.

C. Outline the process of building an application:

- Determine requirements
- Research and analysis
- Design
- Develop
- Test
- Deployment
- Support and maintenance
- Collect feedback

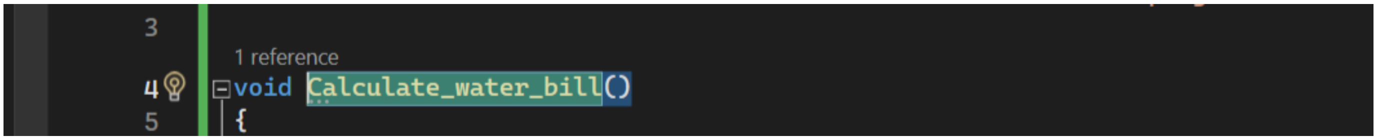
D. Steps taken from coding to execution:

- First, we will print a message line to the screen, informing us about the water bill calculation program.



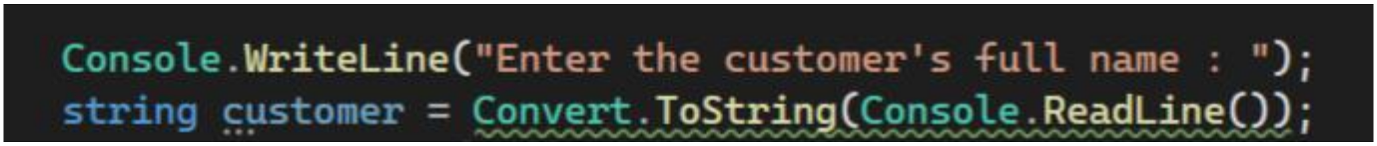
```
// See https://aka.ms/new-console-template for more information
Console.WriteLine("****************** Water bill calculation program ****************");
```

- Define a method named Calculate_water_bill(). This method does not return a value (void), its task is to calculate the water bill.



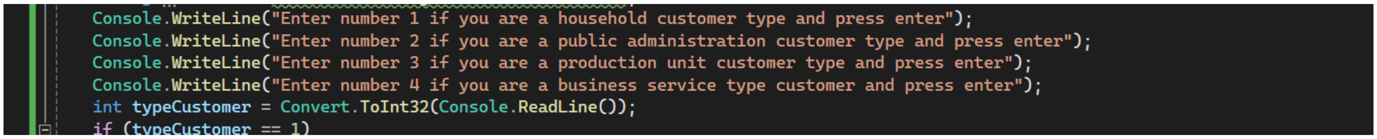
```

1 reference
void Calculate_water_bill()
{
```

- Ask the user to enter the name of the customer and store that value in the variable customer.



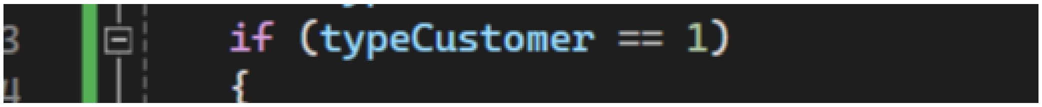
```
Console.WriteLine("Enter the customer's full name : ");
string customer = Convert.ToString(Console.ReadLine());
```

- Require the user to select the type of customer: if it is a household, press number 1, if it is a public administrative agency, press number 2, if it is a manufacturing unit, press number 3, if it is a service enterprise, press number 4 if it is a business service customer type and save the selection in the typeCustomer variable.



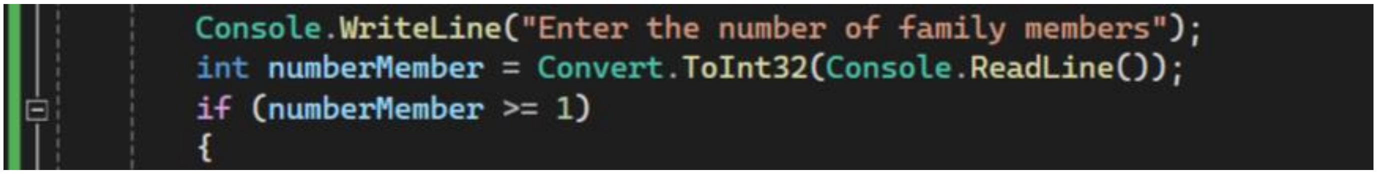
```
Console.WriteLine("Enter number 1 if you are a household customer type and press enter");
Console.WriteLine("Enter number 2 if you are a public administration customer type and press enter");
Console.WriteLine("Enter number 3 if you are a production unit customer type and press enter");
Console.WriteLine("Enter number 4 if you are a business service type customer and press enter");
int typeCustomer = Convert.ToInt32(Console.ReadLine());
if (typeCustomer == 1)
```

- This line checks if the customer type is a home customer. The variable typeCustomer stores the type of customer the user has selected.



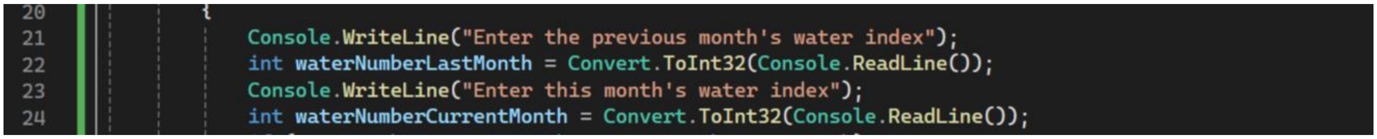
```
if (typeCustomer == 1)
{
```

- This fragment requires the user to enter the number of family members and stores that value in the variable numberMember and checks if the number of family members is greater than or equal to 1.



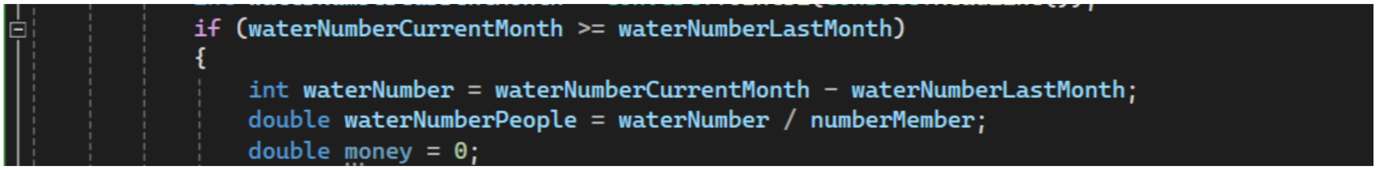
```
Console.WriteLine("Enter the number of family members");
int numberMember = Convert.ToInt32(Console.ReadLine());
if (numberMember >= 1)
{
```

- Requires the user to enter last month's and this month's water meter readings, and store them in the variables waterNumberLastMonth and waterNumberCurrentMonth.



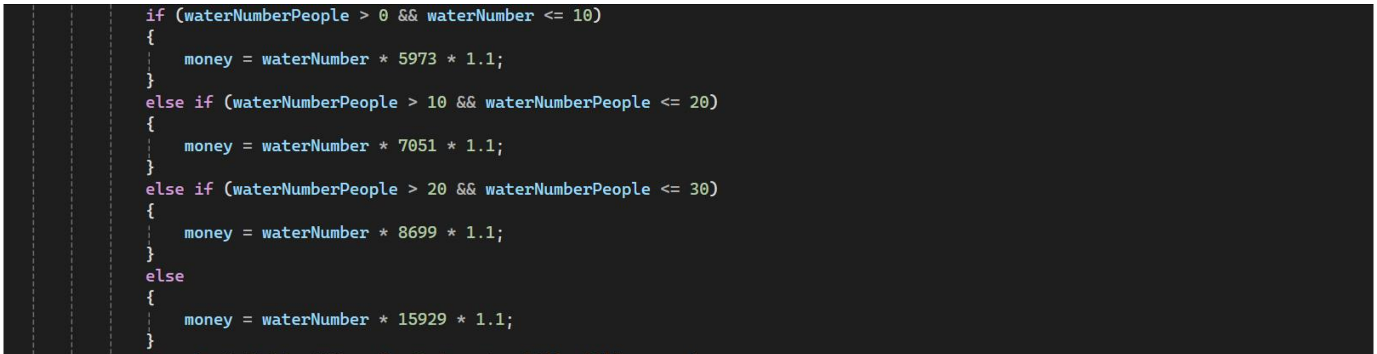
```
{
    Console.WriteLine("Enter the previous month's water index");
    int waterNumberLastMonth = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Enter this month's water index");
    int waterNumberCurrentMonth = Convert.ToInt32(Console.ReadLine());
```

- Check if last month's water index is less than or equal to this month's water index and calculate the amount of water used and the average amount of water per person, then initialize the money variable to calculate the payment amount .



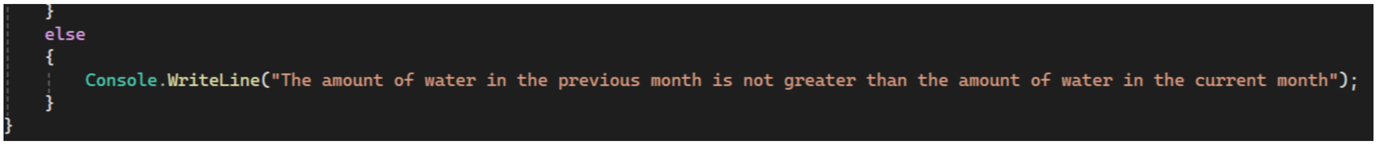
```
if (waterNumberCurrentMonth >= waterNumberLastMonth)
{
    int waterNumber = waterNumberCurrentMonth - waterNumberLastMonth;
    double waterNumberPeople = waterNumber / numberMember;
    double money = 0;
```

- Based on the average amount of water per person, the program calculates the payment amount based on different water prices. Rates are applied according to the amount of water used and print to the screen the amount to pay for the family's water bill based on previous classifications and calculations.



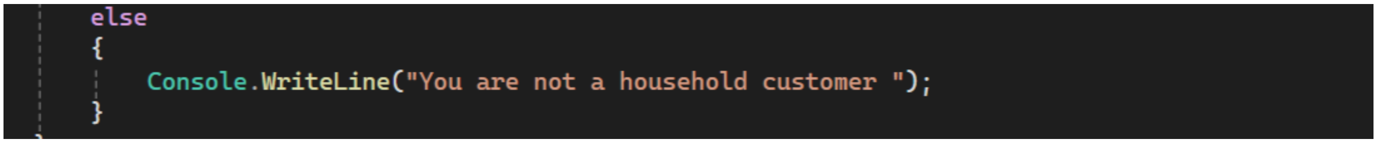
```
if (waterNumberPeople > 0 && waterNumber <= 10)
{
    money = waterNumber * 5973 * 1.1;
}
else if (waterNumberPeople > 10 && waterNumberPeople <= 20)
{
    money = waterNumber * 7051 * 1.1;
}
else if (waterNumberPeople > 20 && waterNumberPeople <= 30)
{
    money = waterNumber * 8699 * 1.1;
}
else
{
    money = waterNumber * 15929 * 1.1;
}
```

- Notice that the number of countries in the previous month is not greater than the number of countries in this month.



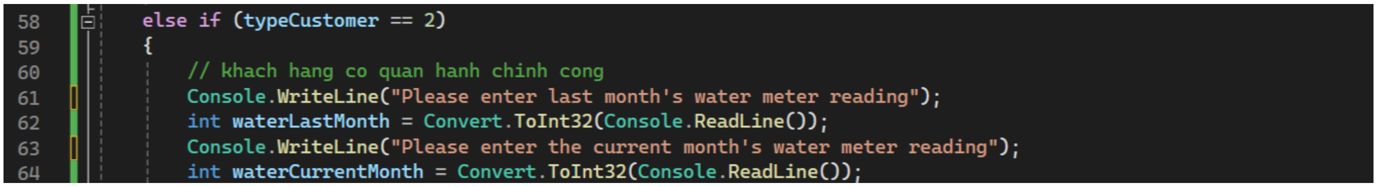
```
    }
    else
    {
        Console.WriteLine("The amount of water in the previous month is not greater than the amount of water in the current month");
    }
}
```

- Notice that the user is not a home customer.



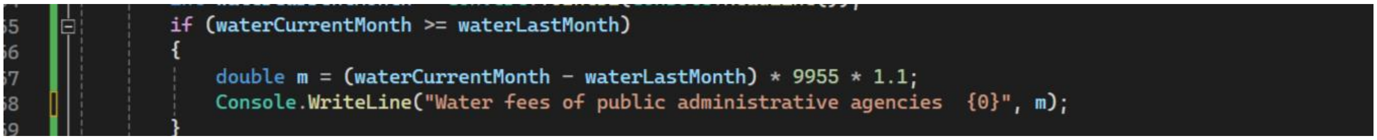
```
        else
        {
            Console.WriteLine("You are not a household customer ");
        }
```

- If the customer type is a public administration, the program requires the user to enter the water meter readings of the previous month and this month.



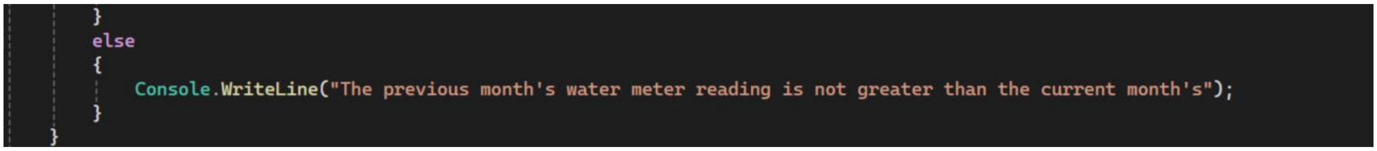
```
            else if (typeCustomer == 2)
            {
                // khach hang co quan hanh chinh cong
                Console.WriteLine("Please enter last month's water meter reading");
                int waterLastMonth = Convert.ToInt32(Console.ReadLine());
                Console.WriteLine("Please enter the current month's water meter reading");
                int waterCurrentMonth = Convert.ToInt32(Console.ReadLine());
```

- If the previous month's water index is less than or equal to this month's water index, the program calculates the amount to be paid based on the amount of water used, with a water price of 9955 VND/m³ and an additional 10% fee.



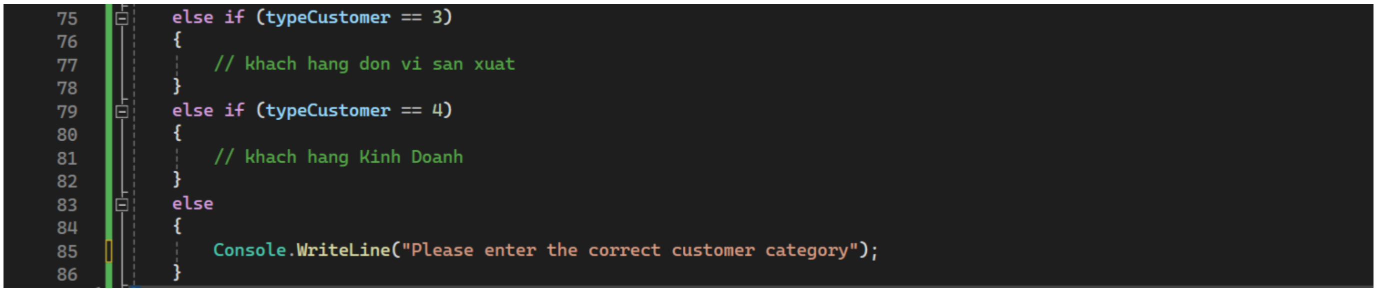
```
                if (waterCurrentMonth >= waterLastMonth)
                {
                    double m = (waterCurrentMonth - waterLastMonth) * 9955 * 1.1;
                    Console.WriteLine("Water fees of public administrative agencies  {0}", m);
                }
```

- If last month's water index is greater than this month's water index, the program reports that last month's water index is not greater than this month's water index.



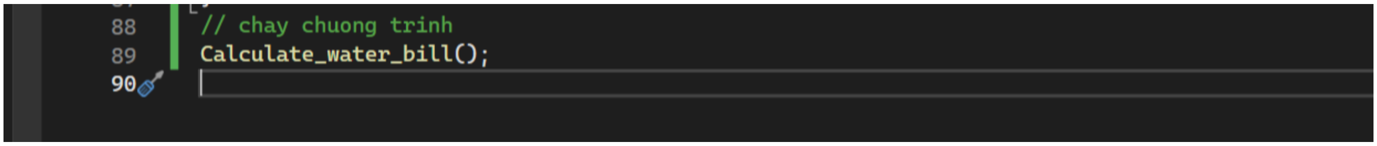
```
        }
        else
        {
            Console.WriteLine("The previous month's water meter reading is not greater than the current month's");
        }
    }
```

- For other types of customers such as production and business units, there is currently no specific treatment for them. The user will be asked to re-enter the customer type if the selection is invalid.



```
else if (typeCustomer == 3)
{
    // khach hang don vi san xuat
}
else if (typeCustomer == 4)
{
    // khach hang Kinh Doanh
}
else
{
    Console.WriteLine("Please enter the correct customer category");
}
```

- Finally, after processing the water bill for the customer type, the program calls the Calculate_water_bill() method to run the program.



```
// chay chuong trinh
Calculate_water_bill();
```

E. The coding process in this example includes several steps and potential challenges:

- Identify requirements: First of all, defining exact requirements is extremely important. In this case, the requirement is to write a program that calculates water bills for different types of customers based on input parameters such as water index.
- Logic design: After understanding the requirements, you need to design the logic of the program. This includes determining the steps required to calculate water bills for each type of customer.
- Choosing data structures and algorithms: During the design process, you need to choose the appropriate data structure and algorithm to solve the problem. In this case, the program uses conditional statements to determine the customer type and calculate the invoice based on the input data.
- Coding: Once you've finished designing, you'll start coding. This process involves using a specific programming language to implement the designed logic into source code.
- Testing and debugging: After writing the code, you need to perform testing to ensure that the program works as expected and handles special cases properly. This includes checking for invalid input values, handling exceptions, and ensuring the accuracy of calculation results.

F. Potential challenges during coding may include:

- Ensuring Logic Correctness: Ensuring that a program's logic works properly and calculates results correctly is an important challenge.
- Handling invalid input: This is especially important when the program interacts with the user. You need to handle situations where users enter invalid or malformed data.
- Performance and optimization: In programs that are large or require processing large amounts of data, performance can become a challenge. Optimizing code to reduce execution time and system resources is an important factor.
- Maintainability and extensibility: Ensuring that code can be easily maintained and expanded in the future is an important factor in software development.

III. CONCLUTION

- In short, the water bill calculation program that I used the C# language to write is a tool to help calculate water bills for households, businesses, public administrative agencies, and production units. The exporter can calculate the amount of water more easily.

IV. REFERENCES

- CodeLearn (n.d.). *Thuật toán là gì? Học thuật toán làm quái gì?* [online] codelearn.io. Available at: https://codelearn.io/sharing/thuat-toan-la-gi-hoc-thuat-toan-lam-quai-gi.