**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** | TRẦN VĂN MINH VŨ | **Student ID** | GCD18517 |
| **Class** | GCD 0825 | **Assessor name** |  |
| **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** |  |

**Grading grid**

|  |  |  |
| --- | --- | --- |
| P1 | M1 | D1 |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **❒ Summative Feedback: ❒ Resubmission Feedback:** | | |
| **Grade:** | **Assessor Signature:** | **Date:** |
| **Lecturer Signature:** | | |

**P1 Provide a definition of what an algorithm is and outline the process in building an application.**

Define Algorithm:

The algorithm is a finite sequence of operations arranged in a certain sequence so that after performing that sequence of operations, the input we have is the output to be searched.

-Algorithm nature:

+Accuracy: to ensure that the calculation results or operations performed by the computer are correct.

Example: I have an algorithm

**Main();**

**Int sum,a,b,;**

**sum=a+b;**

When I replace a = 1, b =1. So Sum = 2. The algorithm exactly.

+Clarity: Algorithms must be expressed by transparent statements; The statements are arranged in a certain order.

Algorithm:

Steps:

**(1)Main()**

**(2)Int sum,a,b;**

**(3)a=Convert.ToInt32(Console.ReadLine());**

**(4)b=Convert.ToInt32(Console.ReadLine());**

**(5)Sum a+b;**

**(6)Console.Write(“Sum ab is:”+sum);**

If in the algorithm I change 6 on 5 the algorithm will **Error**. Because 6 is step print result of algorithm so it can’t on 5 is step calculator of algorithm

+Objectivity: An algorithm, though written by many people on multiple computers, must produce the same result.

Main()

Int sum,a,b;

Scanf(%d&a);

Scanf(%d&b);

sum=a+b;

print(“Result sum:%d”,&sum);

when algorithm running on other programming languages still gives results.

+Popularity: Algorithms not only apply to a certain problem but can apply to a class of problems with similar inputs.

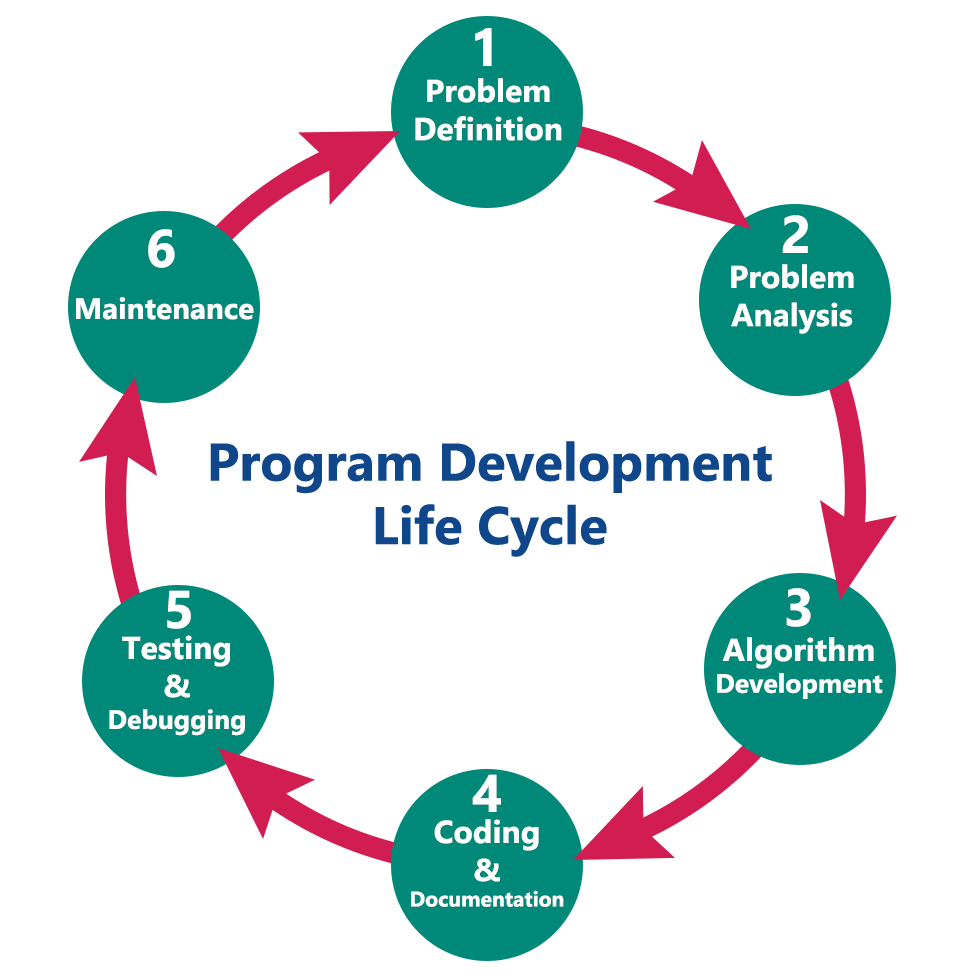
As the above example, if I replace a and b with another number, it still produces a correct result of a + b.

+Ending: The algorithm must include a finite number of computational steps.

**Console.Write(“Sum ab is:”+sum);**

This is step end algorithm.

**Steps to Writing a Program**



**Step1**: Problem Definition

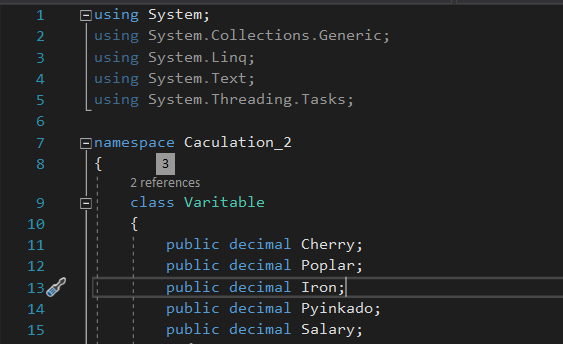
- Calculate the total amount of company earned and the company's interest.

**Step2**: Design a solution

-Input:

+The woods:

Cherry,Poplar,Iron,Pyinkado,salary.



+All value will be decimal

+For each input item, valid range of values that it may assume is ±1.0 × 10e−28 -> ±7.9 × 10e28

+These values are limited to low processing speeds. Because the domain is big

-Output:

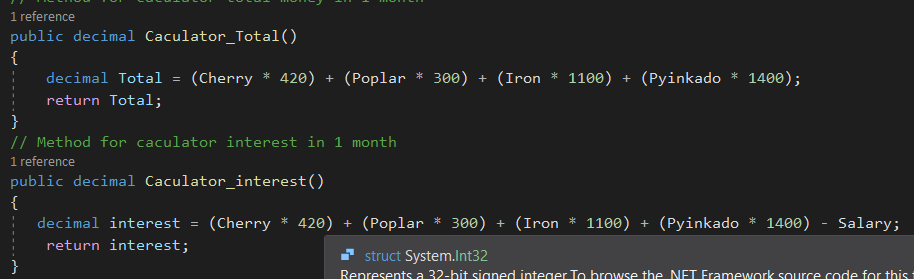


+Console print out value Total and Interest

+The format of these values is decimal

-Progress:

Calculator total and interest are needed



-Flowchart:

True

If input= Y or Yes

Read input

Start (Class valid)

Start (Class variable)

RT

False

Initialize input=string

Function Menu

EXIT

If input= another key

Output enter to continue

Output value interest

Output value total

User input variable to calculator (cherry,poplar,iron,pynikado,salary

Create instances VARI with class variable

Function Main

Output menu

(Cherry \* 420) + (Poplar \* 300) + (Iron \* 1100) + (Pyinkado \* 1400) - Salary;

Function calculator interest

(Cherry \* 420) + (Poplar \* 300) + (Iron \* 1100) + (Pyinkado \* 1400);

Function calculator total

Dim Salary

Dim pyinkado

Dim Iron

Dim Poplar

Dim Cherry

-Write Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Caculation\_2

{

class Variable

{

// Declare attributes

public decimal Cherry;

public decimal Poplar;

public decimal Iron;

public decimal Pyinkado;

public decimal Salary;

// Method for calculator total money in 1 month

public decimal Calculator\_Total()

{

decimal Total = (Cherry \* 420) + (Poplar \* 300) + (Iron \* 1100) + (Pyinkado \* 1400);

return Total;

}

// Method for calculator interest in 1 month

public decimal Calculator\_interest()

{

decimal interest = (Cherry \* 420) + (Poplar \* 300) + (Iron \* 1100) + (Pyinkado \* 1400) - Salary;

return interest;

}

}

public class Valid

{

//create menu items for company

static void Menu()

{

Console.Write("The company sells the following wooden box items\n");

Console.Write("1.Cherry-Price:420USD/m3\n");

Console.Write("2.Poplar-Price:300USD/m3\n");

Console.Write("3.Iron-Price Laos:1100USD/m3\n");

Console.Write("4.Pyinkado Laos:1400USD/m3\n");

}

public static void Main(string[] args)

{

RT:

// calling function menu

Menu();

// Create instances VARI with class Vaitable

Variable VARI = new Variable();

// create method to user input

Console.Write("Amount Cherry-Price : "); VARI.Cherry = Convert.ToDecimal(Console.ReadLine());

Console.Write("Amount Poplar-Price: "); VARI.Poplar = Convert.ToDecimal(Console.ReadLine());

Console.Write("Amount Iron-Price: "); VARI.Iron = Convert.ToDecimal(Console.ReadLine());

Console.Write("Amount Pyinkado: "); VARI.Pyinkado = Convert.ToDecimal(Console.ReadLine());

Console.Write("Salary Employeer: "); VARI.Salary = Convert.ToDecimal(Console.ReadLine());

//calling method Calculator\_Total of VARI instances and print out value

Console.WriteLine("Total: " + VARI.Calculator\_Total());

//calling method Caculator\_interest of VARI instances and print out value

Console.WriteLine("interest:" + VARI.Calculator\_interest());

Console.WriteLine("ENTER TO CONTINUE");

Console.ReadKey();

//Create input again and if input yes will go back program,another key will exit

string input;

Console.Write("DO YOU WANT RETURN: INPUT 'Y' OR 'Yes' To RETURN,Another key to Exit: ");

input = Console.ReadLine();

if (input == "Y"||input=="Yes")

{

goto RT;

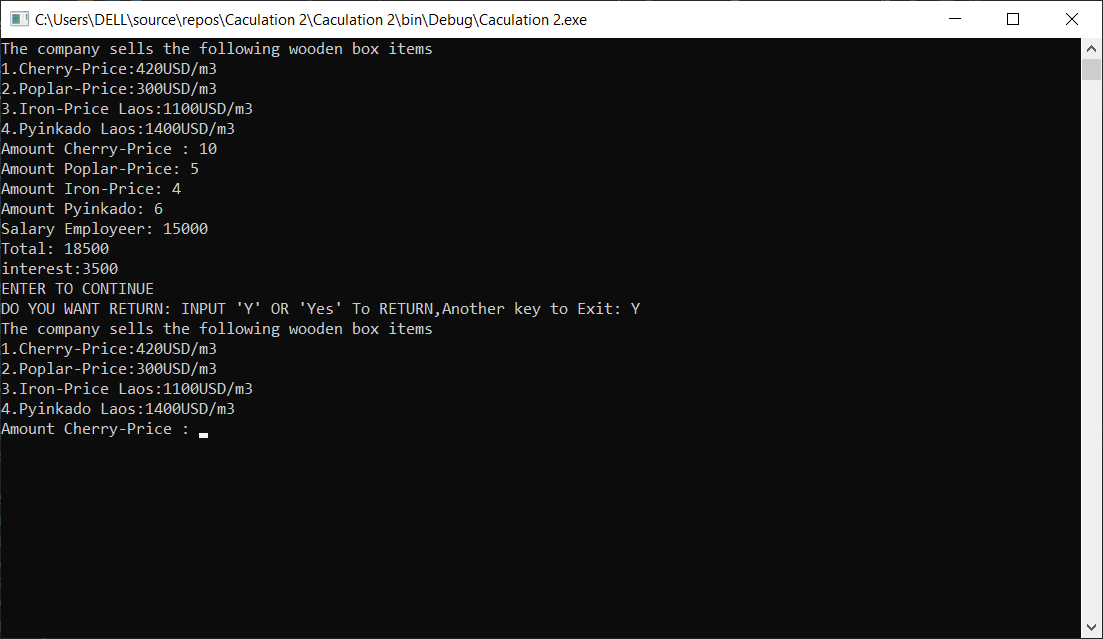
}

}

}

}

-Test with real-world users:



**Distinction:**

D1 Examine the implementation of an algorithm in a suitable language. Evaluate the relationship between the written algorithm and the code variant.

Algorithms have been implemented on c # languages.

-Evaluate the relationship between the written algorithm and the code variant:

+Algorithm:

The algorithm is a finite sequence of operations arranged in a certain sequence so that after performing that sequence of operations, the input we have is the output to be searched.

+Code:

The code is a string of computer algorithms and the code is written in a programming language so that the computer can understand and execute the algorithm on the computer.

Code variant is a way for expressing algorithms on computers, in programming languages. Code variant relatively uncomplicated but if the algorithm is complex then a will be written strictly to resemble the algorithm and display the correct results of the algorithm.