**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** |  | **Student ID** |  |
| **Class** |  | **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:** | | |

|  |
| --- |
| **Assignment Brief and Guidance:** |
| **Assignment scenario**  You have applied for a post as a trainee with a software development company and have been invited for an interview. You have been asked to demonstrate your problem solving and basic programming skills. To do this you have to prepare a report on using algorithms to solve problems.  You need to explain, using examples, how algorithms are used to solve simple business problems and the steps needed to be followed to produce a working program solution. You should make clear your assumption about your program. The problems to be solved will involve basic procedural programming instructions - sequence instructions (input, output and assignment statements), loops, conditional statements. Problems should be analysed and designed by the use of flowchart and demonstrated by the use of modules (procedures).  **Tasks:**   * State your simple business problems to be solved. * Analyse the problem and design the solutions by the use of suitable methods. * Demonstrate the compilation and running of a program * Evaluate how the problem is solved from the designed algorithm to the execution program written by a specific programming language. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Learning Outcomes and Assessment Criteria (Assignment 1):** | | |  |
| Learning Outcome | Pass | Merit | Distinction |
| LO1 | **P1** Provide a definition of what an algorithm is and outline the process in building an application. | **M1** Determine the steps taken from writing code to execution. | **D1** Examine the implementation of an algorithm in a suitable language. Evaluate the relationship between the written algorithm and the code variant. |

***LO1*** *Define basic algorithms to carry out an operation and outline the process of programming an application*

# Table of content

Contents

[Table of content 5](#_Toc101981965)

[P1 Provide a definition of what an algorithm is and outline the process in building an application 7](#_Toc101981966)

[Chapter 1 - State your simple business problems to be solved 7](#_Toc101981967)

[1.1 Overview about Algorithm 7](#_Toc101981968)

[1. What is an Algorithm 7](#_Toc101981969)

[-So, There are 3 main features of the algorithm according to its definition: 8](#_Toc101981970)

[An algorithm is a step-by-step description of the solution to a problem 9](#_Toc101981971)

[-Steps in Program Development: 12](#_Toc101981972)

[-Types of algorithms 14](#_Toc101981973)

[-Example an Althorightm 17](#_Toc101981974)

[1.1 Main elements of (use-case diagram) library 19](#_Toc101981975)

[1.2 Represent a small and simple problem 22](#_Toc101981976)

[Represent a small, simple problem and solve this problem 22](#_Toc101981977)

[Chapter 2 - Analyse the problem and design the solutions by the use of suitable methods 23](#_Toc101981978)

[2.1 Analyse the problem 23](#_Toc101981979)

[Analys the problem 23](#_Toc101981980)

[2.2 Flowchart 25](#_Toc101981981)

[Finish the Real code with your Imaging Use-case diagram, flow chart 25](#_Toc101981982)

[Chapter 3 - Demonstrate the compilation and running of a program 26](#_Toc101981983)

[Demonstrate the compilation and running of a program 26](#_Toc101981984)

[-Write Pseudo-code <Code Flow> 28](#_Toc101981985)

[3.1 Introduce how the problem is solved 33](#_Toc101981986)

[● Design a solution 33](#_Toc101981987)

[-1. Introduce how the problem is solved 35](#_Toc101981988)

[3.2 Source code and screen shots of the final result 35](#_Toc101981989)

[-Write Code C#: 35](#_Toc101981990)

[Source code and screenshots of the final result 37](#_Toc101981991)

[Case 1: passed it’s working 38](#_Toc101981992)

[Case 2: Also working 39](#_Toc101981993)

[Case 3: 39](#_Toc101981994)

[Case 4: 40](#_Toc101981995)

[3.3 Explain briefly what is Software Development Life Cycle 41](#_Toc101981996)

[-Steps in Program Development: 42](#_Toc101981997)

[3.4 Explain how the source code is compiled 44](#_Toc101981998)

[3.Explain how to source is complied 44](#_Toc101981999)

[Chapter 4 - Evaluate how the problem is solved from the designed algorithm to the execution program written by a specific programming language 46](#_Toc101982000)

[4.1 Include Test cases 46](#_Toc101982001)

[M1 Determine the steps taken from writing code to execution 46](#_Toc101982002)

[4.2 Evaluate how the problem is solved from the designed algorithm to the execution program written by a specific programming language 48](#_Toc101982003)

[IV. Eveluate how to problem is solved from the designed algorithm to the execution program written by a specific programming laguage 48](#_Toc101982004)

[D1 Examine the implementation of an algorithm in a suitable language. Evaluate the relationship between the written algorithm and the code variant 50](#_Toc101982005)

[REFERENCES 53](#_Toc101982006)

[(https://www.geeksforgeeks.org/linear-search/, n.d.) 53](#_Toc101982007)

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

# Chapter 1 - State your simple business problems to be solved

## 1.1 Overview about Algorithm

### 1. What is an Algorithm

-An algorithm (Programming) can be defined as a step-by-step method, used to solve a problem. By using "A sequence of steps will be performed for a required output from a given input" You can think of a programming algorithm as a formula that describes the exact steps required for a computer to solve problems or achieve goals.

For example: We all have seen recipes - they list the necessary ingredients and a set of steps to make the meal described.

Yes, an algorithm is like that. It only includes what you need to perform the task. It does not include anything unclear, often called vague in computer jargon, that someone reading it may wonder about.

In computer jargon, the word for a formula is a procedure and components are called inputs. Labeling the first step is 'starting' and the last step is 'ending' .Your computer looks at your procedure, follows it to the letter and you can see the result, called the output. A programming algorithm describes how to do something and your computer will do exactly that way all the time. That's right, it will once you convert your algorithm into the language it understands!

It is often used for data processing, calculations and other mathematical and computer related activities. The essential purpose of an algorithm is to get a specific output, An algorithm consists of several continuous steps, An algorithm is used to manipulate data in many different ways, such as inserting a new data item, searching for a specific item or sorting an item. Output appears after the algorithm finishes the whole process.

Basically, all algorithms perform logically while following the steps to get the output to a given input. However, it is important to note that the programming algorithm is not computer code.

It always leads to a solution and tries to be the most effective solution we can think of. It's usually a good idea to number steps, but you don't have to. Instead of numbered steps, some users indent and write it in pseudocode, this is a semi-programming language used to describe the steps in the algorithm. But, we will not use that here because simplicity is the main thing. Others just need to use a diagram also known as a "diagram".

### -So, There are 3 main features of the algorithm according to its definition:

1. The essential purpose of an algorithm is to obtain a specific output,

1. An algorithm consists of several continuous steps,
2. Output appears after the algorithm finishes the whole process.

Basically, all algorithms perform logically while following the steps to get the output to a given input.

### An algorithm is a step-by-step description of the solution to a problem

-An algorithm must be

1. Definite
2. Finite
3. Precise and Effective
4. Implementation independent



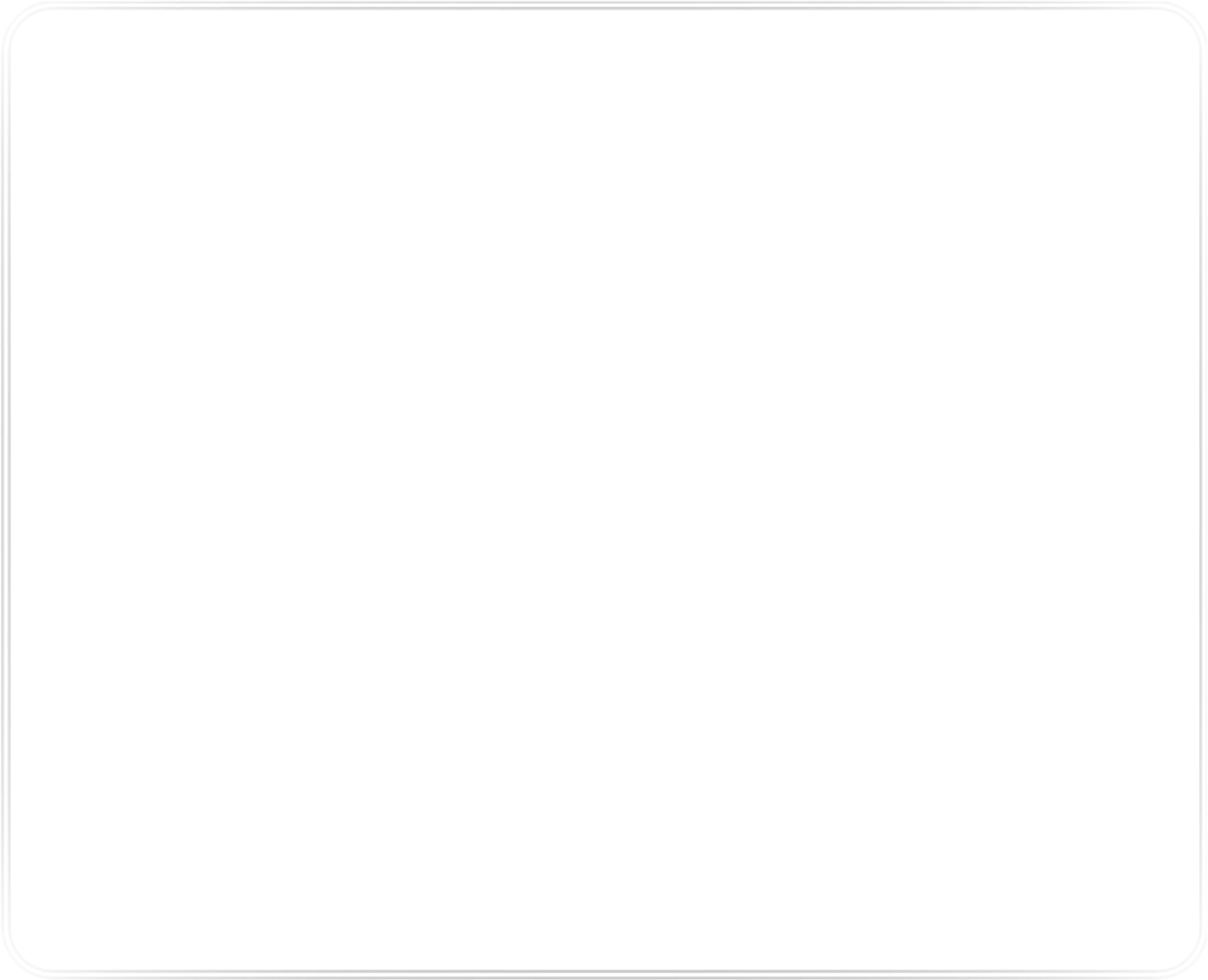
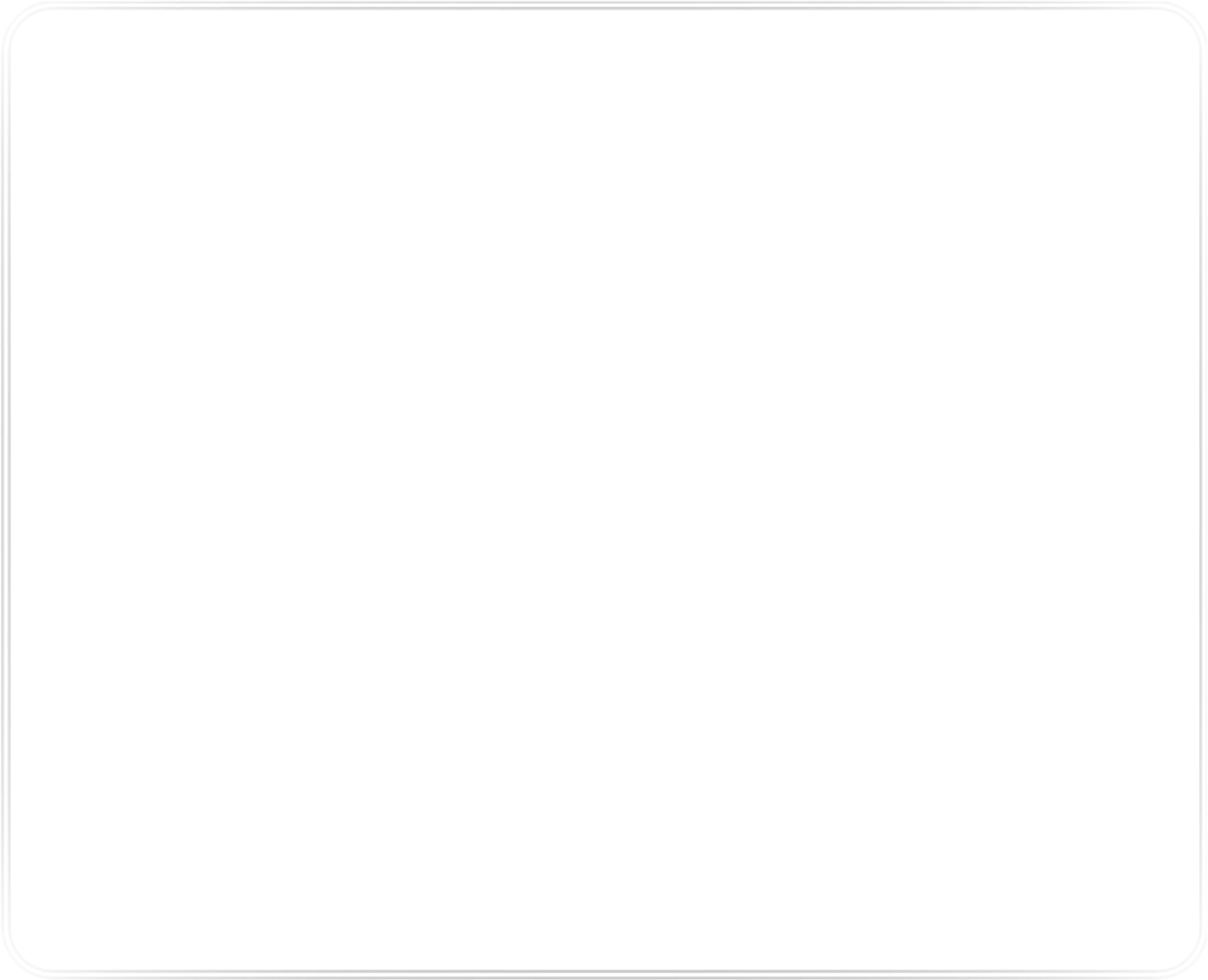
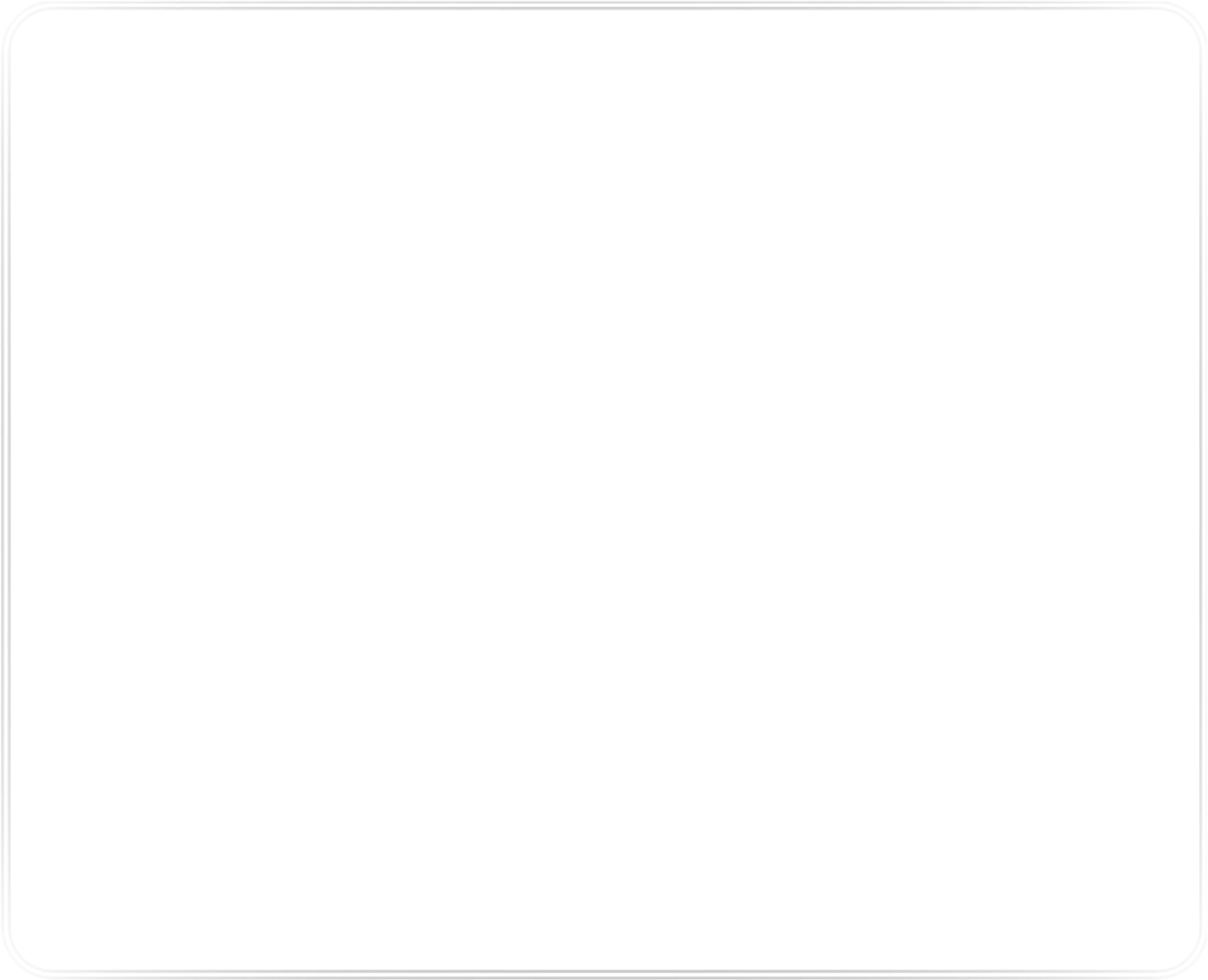
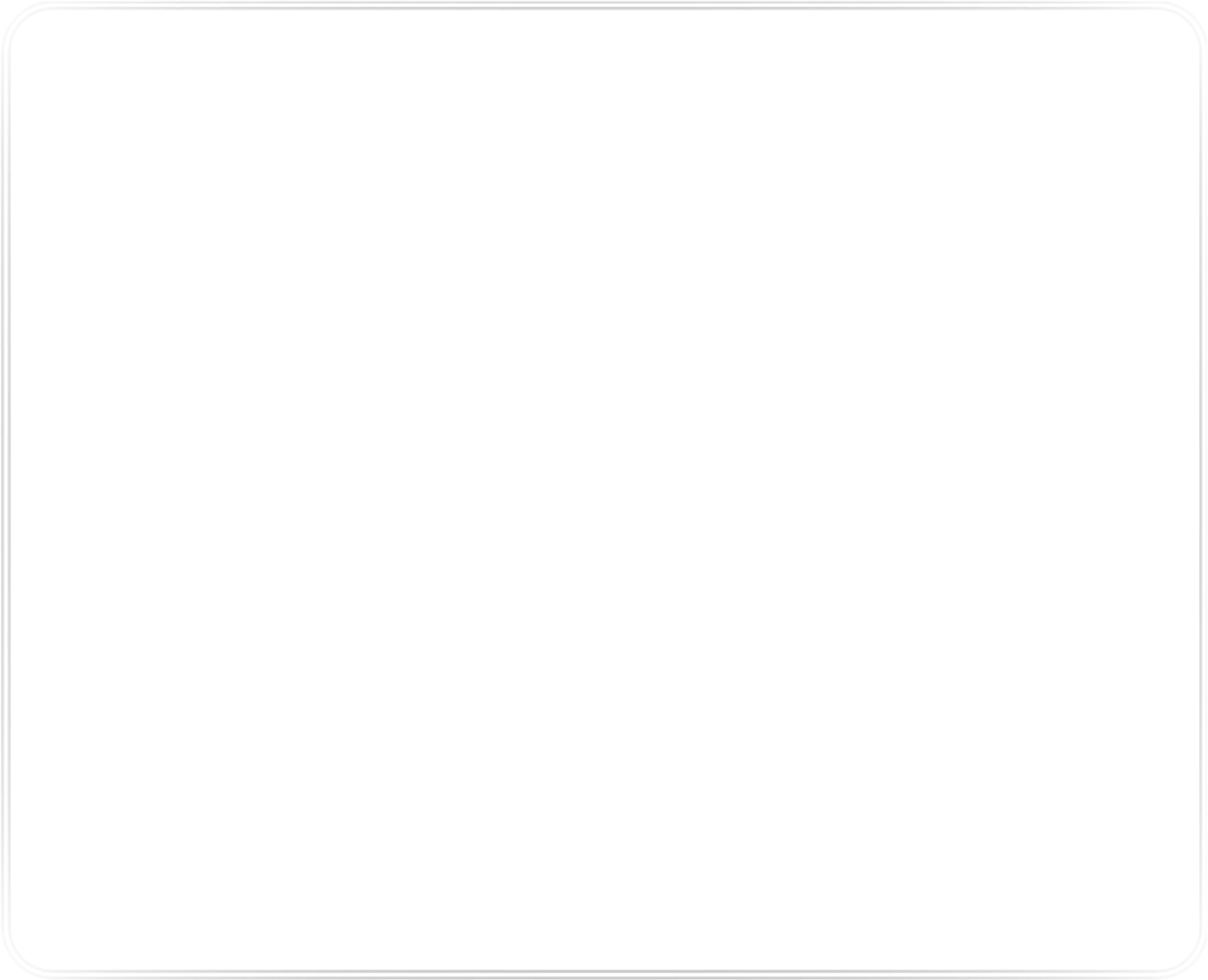
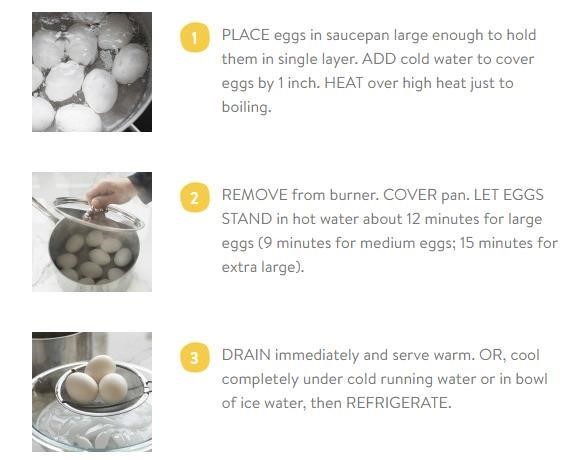
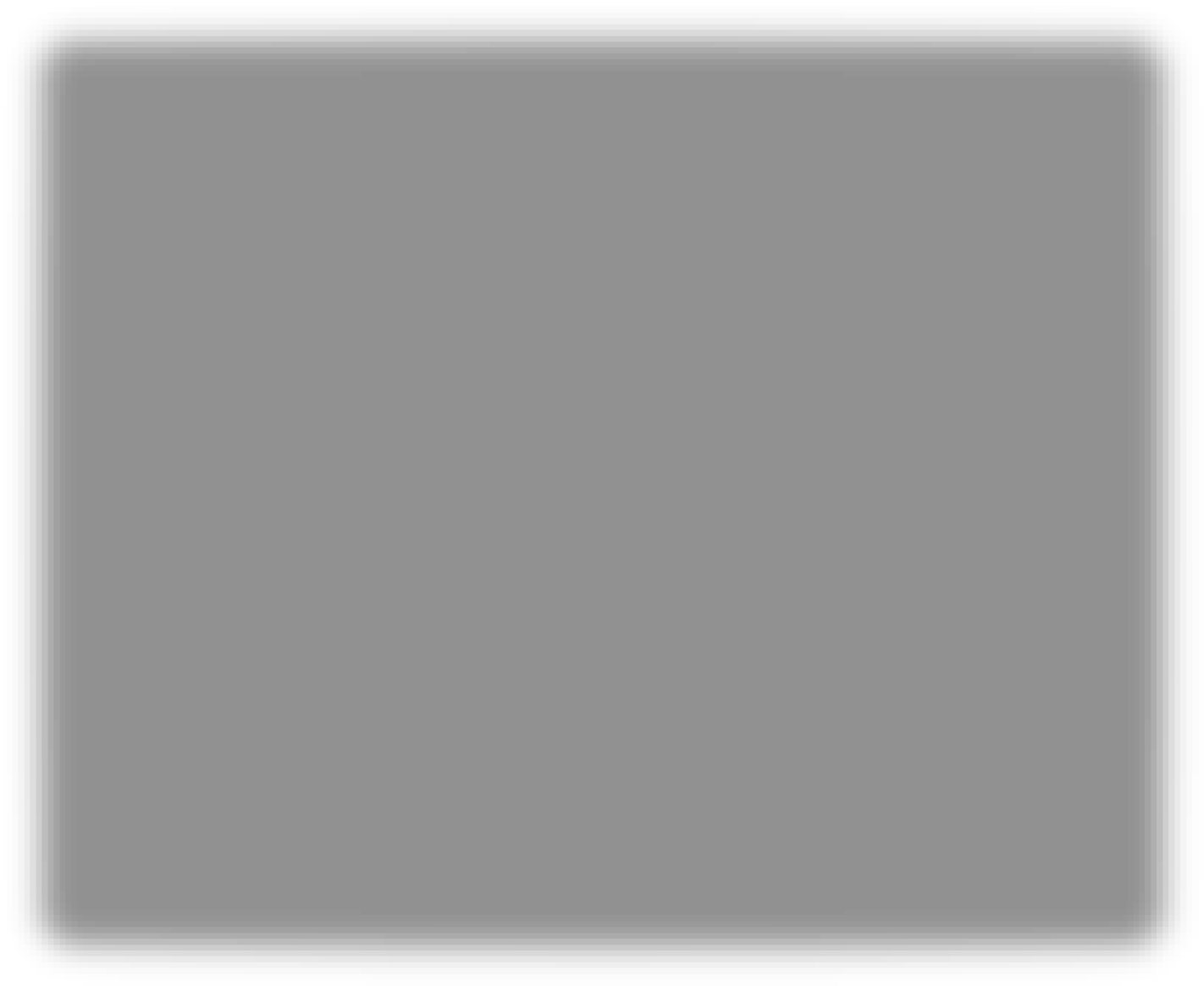
(

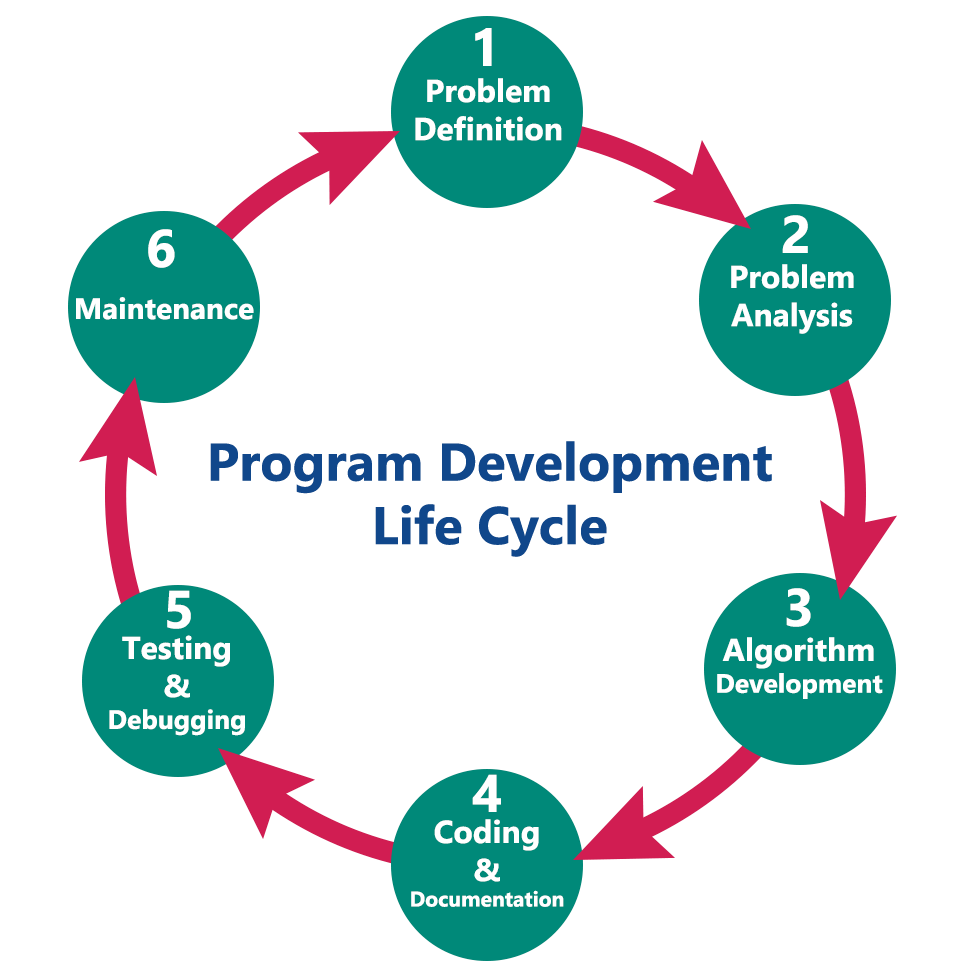


incredibleegg.org, n.d.



)



(btechsmartclass.com, n.d.)

### -Steps in Program Development:

#### In general, the program development lifecycle contains 6 stages, which occur as follows.

1. Problem Definition
2. Problem Analysis
3. Algorithm Development
4. Coding & Documentation
5. Testing & Debugging
6. Maintenance

1. Define the problem

In this stage, this step, we (a programmer) study the problem, determine the problem statement and we decide the boundary of the problem. In this stage, we need to understand to determine the best way to solve these problems and report the problem of what our requirements are, the output of the problem solution. They are defined in this first phase of the program development life cycle.

Researching a problem is also necessary because it helps the programmer decide on the following:

The facts and figures needed to develop the program.

How the program will be designed

In addition, the language in which the program will be most appropriate.

What is the desired output and in what form is necessary, etc.

1. Analyze the problem

In phase 2, we define requirements such as variables, functions, etc. to solve the problem. That means we collect the resources needed to solve the problem identified during the problem identification phase. We also determine the limit of the solution.

1. Development of algorithms

In this stage, we develop a step-by-step process (Algorithm) to solve the problem using the specifications given in the previous period.

The algorithm is a series of steps that must be performed before the programmer starts preparing his program. Programmers design an algorithm to help alternative images possible in a program also.

This stage is very important for program development. That means we write solutions according to each step of the report.

The next step after designing the algorithm is to write a high-level language program. This process is called encryption.

1. Encoding & Documentation

This stage uses programming languages to write or implement practical programming instructions for the steps identified in the previous phase. In this stage, we build a reality program. That means we write a program to solve a given problem in programming languages like C, C, Java, etc.

1. Check and debug

During this stage, the program execution process for finding errors or errors is called test execution. We check to see if the code written in the previous step solved the specified problem.

Debugging is a process of detecting, locating and fixing errors in the program. It is done by running the program multiple times.

That means we check the program to see if it solves the problem for different input data values. We also check to see if it provides the desired output. It helps a programmer test the logic of the program. It also ensures that the program is error-free and workable.

1. Maintenance

When the program is completed, its document is prepared. The final document is provided to the user. It guides users on how to use the program as efficiently as possible.

In this stage, the program is actively used by users. If any improvements are found during this period, all stages will be repeated again to make improvements. That means that in this stage, the solution (program) is used by the end user. If users encounter any problems or want to have any enhancements, we need to repeat all stages from the beginning, so that the problem is solved or enhanced is added.

Moreover, another purpose of having documents is to allow other programmers to modify the code if necessary. Documentation should also be done in every step of the program development process

### -Types of algorithms

Algorithms can be categorized into 3 types based on their structure:

1. Sequence: this type of algorithm is characterized with a series of steps and each step will be performed in turn.
2. Branching (Condition statement): this type of algorithm is represented by "if-then" issues. If a condition is true, the output will be A, if the condition is false, the output will be B. This type of algorithm is also called "choice type".

Structure: If- Else

Code: Increment and Decrement operators; Switch statement

1. Loop statement: for this type, the process can be repeated in a certain condition. It is represented by "while" and "for" issues. But make sure the process ends after some conditional loops. This type of algorithm is also called "repeat type".

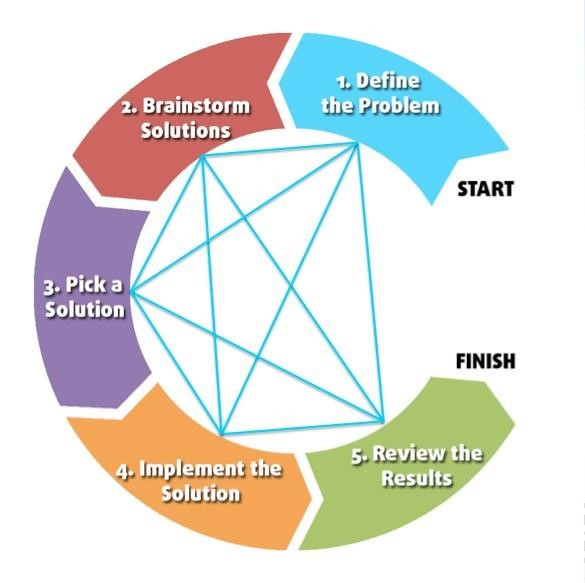
Structure: Do- While

General syntax: "break" for loop to terminate

* Analyzing or Defining the Problem:

Defining a problem by doing a preliminary investigation it is called as Program Analysis

* Tasks for defining a problem

(TARVIN, n.d.)

#### For define a problem, we need:

1. The input ( Determine the outcome of your code)
2. The Processing (Brainstorm Solution, decide on a starting point, find the ending point for Define the problem decide on a starting point for Pick a

Solution and after that, we implement the solution by list the steps from start to finish and Determine how you will accomplish each step)

1. The Output (Review the results)

### -Example an Althorightm

#### Searching Althorightm

Linear Search:

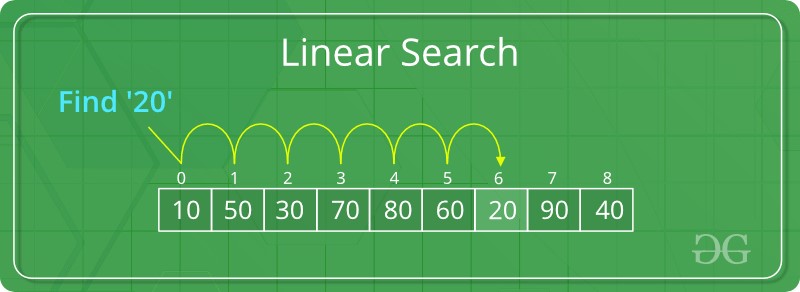
The linear search method is a very basic sequential search technique.

We just traverse the list in its entirety, matching each element to the object whose position must be determined. where do we begin?

A sequential search is performed on all things one by one in this sort of search. Begin at one end of the list and work your way down until you find the ingredient you're looking for. It is the most basic search algorithm. Every item is verified, and if a match is discovered, that item is returned; if not, the search continues until the data collection is complete. It means starting from one end and checking every element in the list until the required element is discovered; if it is, the algorithm returns the item's location; otherwise, it returns NULL.

It is the most basic search algorithm.

Linear search is rarely utilized in practice since other search algorithms, such as the binary search algorithm and hash tables, allow for substantially quicker searching.

(https://www.geeksforgeeks.org/linear-search/, n.d.)

#### - How to Write an Algorithm in Programming Language

Easy way to understand the Algorithm is Make yourself a FlowChart or Use-case diagram for Outside-view (Of System the Algorithm) Design for a path solution in code to solve a problem It will make :

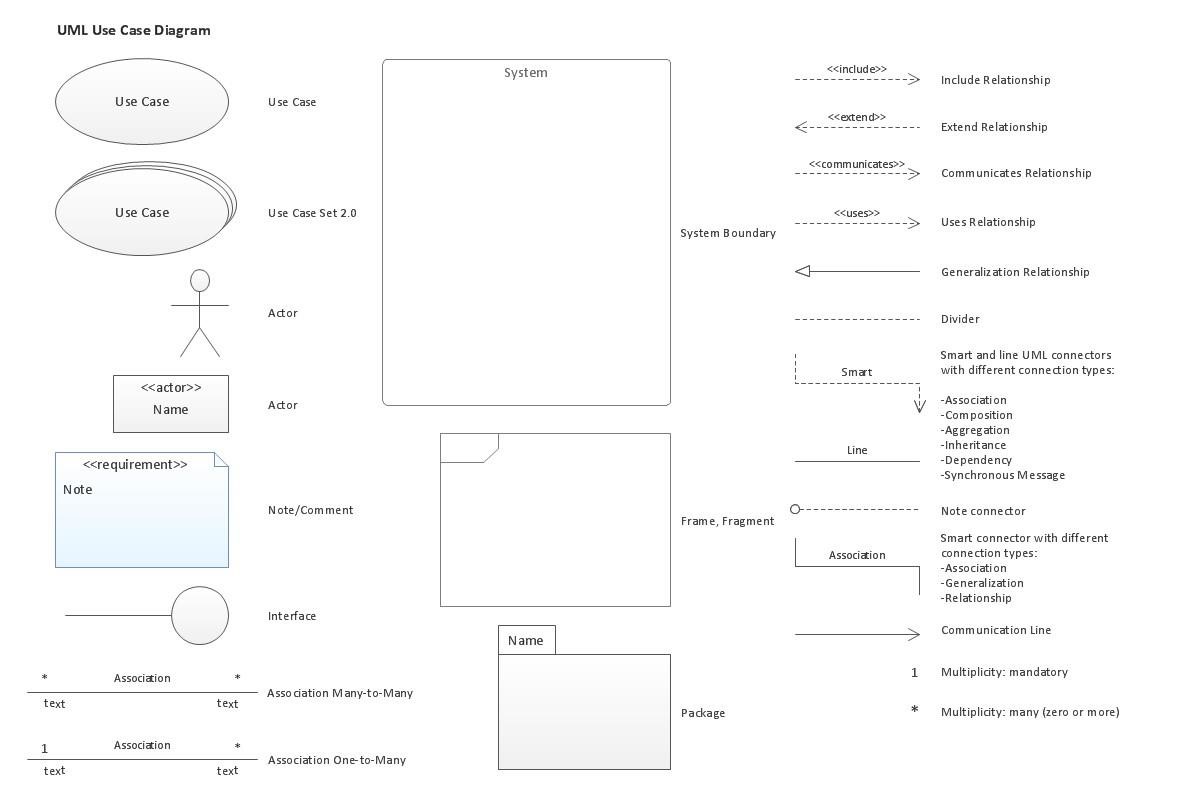
o Easier to read and understand later

o Less of bugs and errors o Easier to extend to add new features

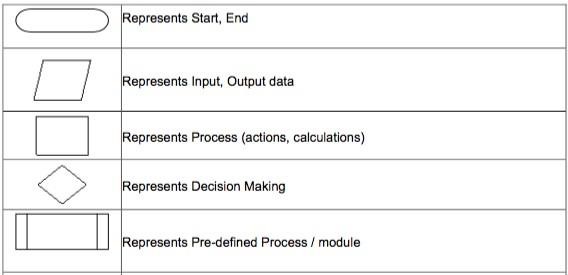
o Easier to program in the first place

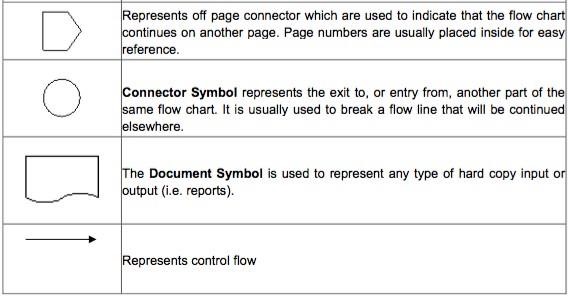
## 1.1 Main elements of (use-case diagram) library

(uml-design-elements-use-case, n.d.)



FlowChart Library

(Slide, n.d.)



## 1.2 Represent a small and simple problem

### Represent a small, simple problem and solve this problem

I'm having trouble creating a selection table to answer the problem "calculate a person's salary and hours in a firm." I'm not sure how to construct a menu selection, find, and save it.

To enter computation data, print a bulletin board.

Make a decision on the following:

-the amount of hours worked

-The entire sum will be paid.

-Penalties for lateness Profit per month

Finally, I'll use the method to solve the problem, which will allow me to divide it down into separate scenarios, making coding easier.

# Chapter 2 - Analyse the problem and design the solutions by the use of suitable methods

## 2.1 Analyse the problem

### Analys the problem

-My first part is to enter the coefficients with case switch choice to show which one to choose

In this sense, I distinguish between the following scenarios:

Input:

Input variables:

1. Work days = wd
2. Leave days = ld
3. Hours per day =hpd
4. Latency hours =lh

5. Paid per hours =pph

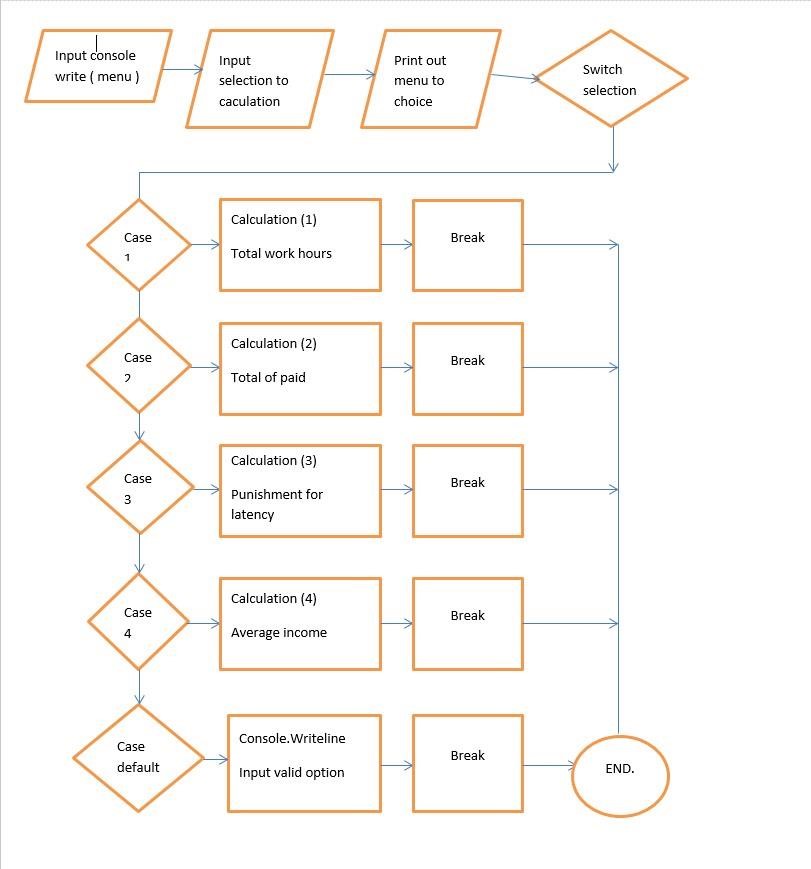
Caculate:

1. Total of work hours/ month = wd \* hpd
2. Total of paid/ month = wd \* hpd \* pph
3. Punishment for latency = lh \* pph
4. Average income/ month of employer = (wd \* hpd \* pph - lh \* pph) / (wd + ld) 5. Exit Programe

Output: Show result

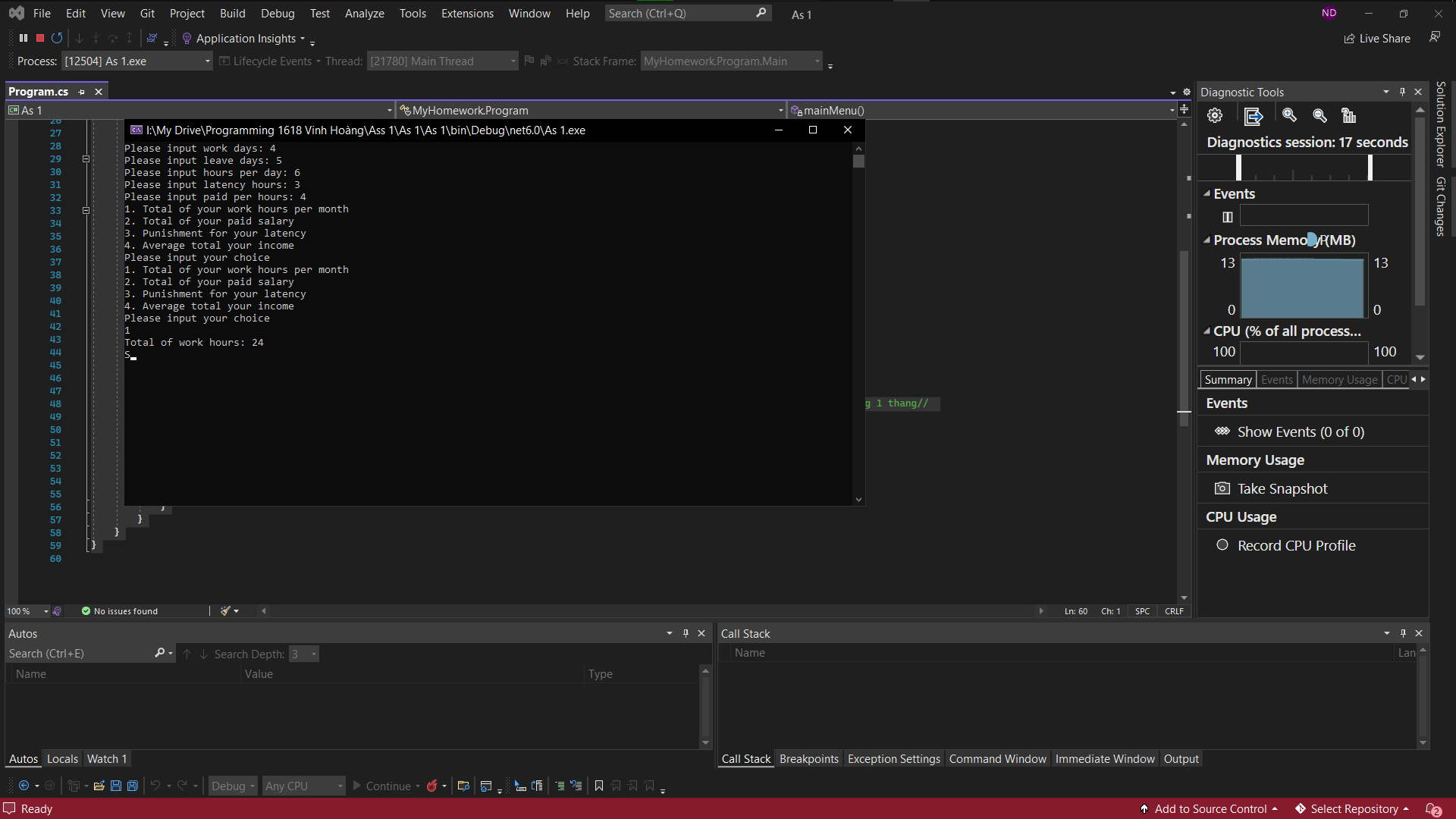
## 2.2 Flowchart

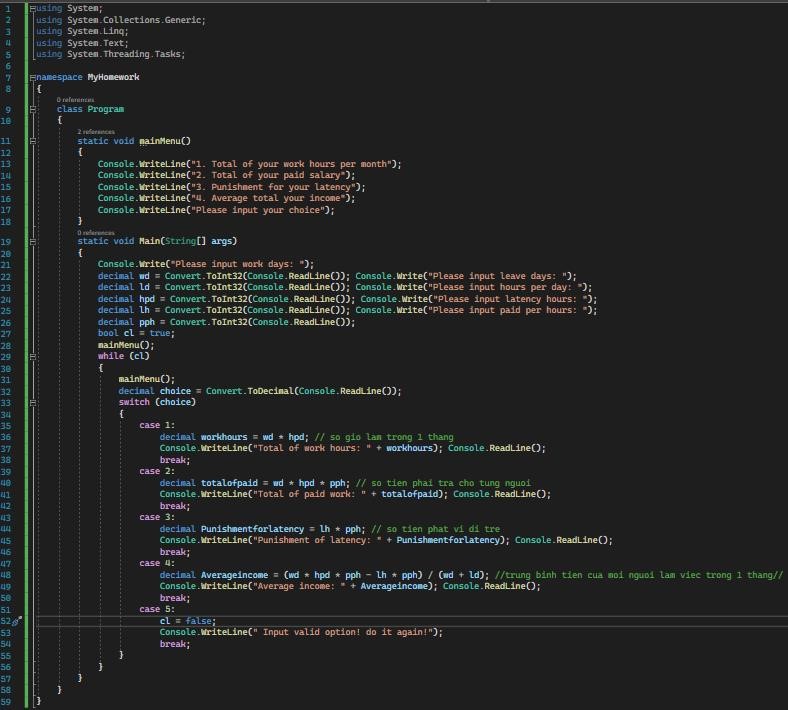
### Finish the Real code with your Imaging Use-case diagram, flow chart



# Chapter 3 - Demonstrate the compilation and running of a program

## Demonstrate the compilation and running of a program

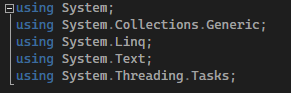




### -Write Pseudo-code <Code Flow>

#### // Library declaration

Khai báo thư viện



#### // Name Space, Class



NameSpace literally translates to the namespace. NameSpace can be illustrated as computers in a company, the data contained in a computer can be easily accessed, if different computers, they must be connected to the internal network, and the link can be read. Similarly, in a small project, only one NameSpace should be used for easy access, if the project is large, it is necessary to allocate many NameSpaces to manage. When we reference dynamic libraries DLL, we refer to other namespace domains.

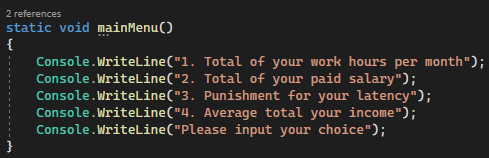


Classes are classes that contain procedures, parameters, methods, .. in general, contain everything. If NameSpace is a computer, Class is a hard drive. A Computer has many hard drives, each of which has a lot of data.

#### //Create menu, Press Input:

Tạo Menu bao gồm

Nhập Input:



In Console Console.WriteLine("1. Total of your work hours per month");

In Console Console.WriteLine("2. Total of your paid salary");

In Console Console.WriteLine("3. Punishment for your latency");

In Console Console.WriteLine("4. Average total your income");

#### // Create Console Print Notifications: About Your Choices

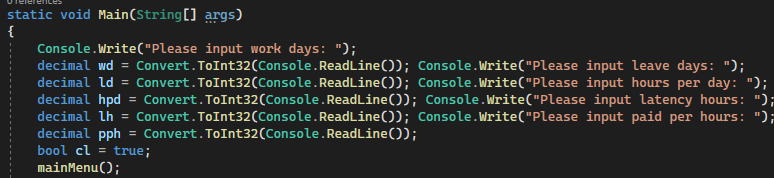
Tạo thông báo in trên Console: Về lựa chọn của bạn



Console.WriteLine("Please input your choice");

Sau đó nhập dữ liệu, người dùng đã bấm vào máy

#### // Then enter the data- that input into Computer, the user pressed on the machine



Console.Write("Please input work days: ");

decimal wd = Convert.ToInt32(Console.ReadLine());

Console.Write("Please input leave days: ");

decimal ld = Convert.ToInt32(Console.ReadLine());

Console.Write("Please input hours per day: ");

decimal hpd = Convert.ToInt32(Console.ReadLine());

Console.Write("Please input latency hours: ");

decimal lh = Convert.ToInt32(Console.ReadLine());

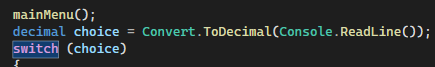
Console.Write("Please input paid per hours: ");

decimal pph = Convert.ToInt32(Console.ReadLine());

bool cl = true;

#### // Generate output request results

Tạo kết quả về yêu cầu xuất ra



\*Case 1: Số giờ làm trong 1 tháng

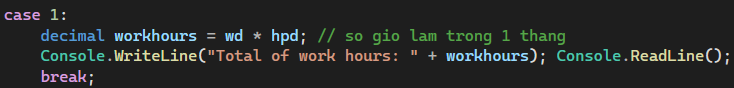
#### //Case 1: Number of hours worked in 1 month

decimal workhours = wd \* hpd; // so gio lam trong 1 thang

Console.WriteLine("Total of work hours: " + workhours);

Console.ReadLine();

break;



\*Case 2: Số tiền phải trả cho từng người

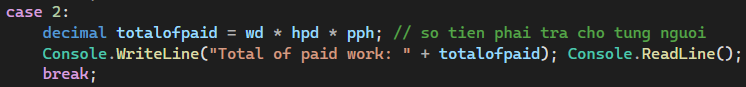
#### // Amount to be paid per person

decimal totalofpaid = wd \* hpd \* pph; // so tien phai tra cho tung nguoi

Console.WriteLine("Total of paid work: " + totalofpaid);

Console.ReadLine();

break;



\*Case 3: Số phạt bị phạt vì đi trễ

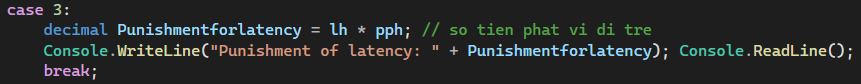
#### // Number of fines for being late

decimal Punishmentforlatency = lh \* pph; // so tien phat vi di tre

Console.WriteLine("Punishment of latency: " + Punishmentforlatency);

Console.ReadLine();

break;



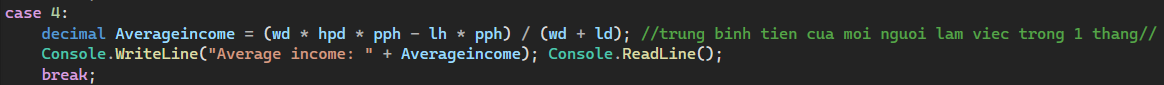
\*Case 4: Trung bình tiền của mỗi người làm việc trong 1 tháng

#### // Average salary of each person working in 1 month

decimal Averageincome = (wd \* hpd \* pph - lh \* pph) / (wd + ld); //trung binh tien cua moi nguoi lam viec trong 1 thang Console.WriteLine("Average income: " + Averageincome);

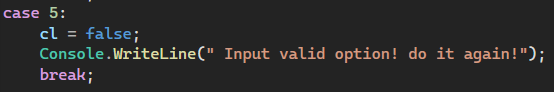
Console.ReadLine();

break;



Nếu sai, thì đưa thông báo yêu cầu nhập lại, vì thông tin nhập sai

#### // If information wrong, then give a message asking to re-enter, because the information entered is wrong



\*Case 5:

Nếu sai

break;

default:

Console.WriteLine(" Input valid option! do it again!");

break;

#### // End the program by pressing any key

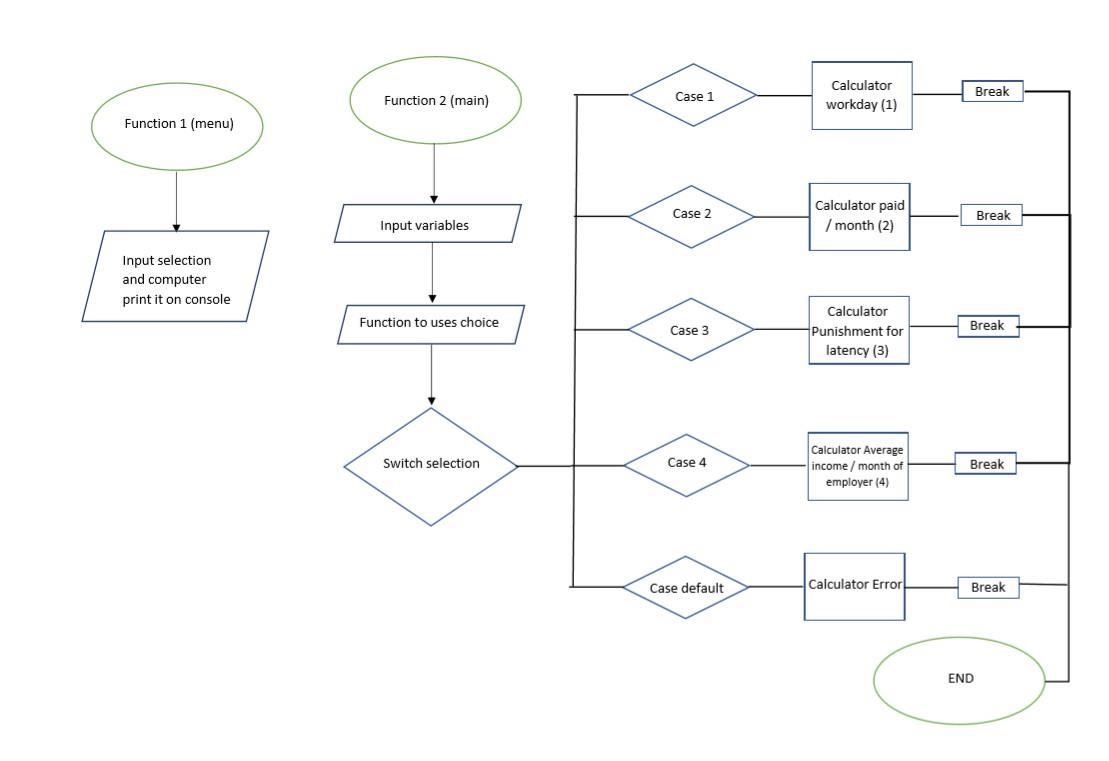
Kết thúc chương trình bằng cách bấm phím bất kỳ

## 3.1 Introduce how the problem is solved

### ● Design a solution

-By my experience my self (With talking)

1. Create Input (Input information)
2. The Processing (Code Algorithms)
3. Create Ouput (Done solve problem) -Draw My Use-case diagram, FlowChart (On my Mind):



### -1. Introduce how the problem is solved

I make a small and simple program. Then I will input 5 coefficients, then the program will assist me in producing the desired output. My little program was

written in the C# programming language.

1.Work days 10

2.Leave days 1

3.Hours per day 10

4.Latency hours 1

5.Paid per hours 500

## 3.2 Source code and screen shots of the final result

### -Write Code C#:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyHomework

{

class Program

{

static void mainMenu()

{

Console.WriteLine("1. Total of your work hours per month");

Console.WriteLine("2. Total of your paid salary");

Console.WriteLine("3. Punishment for your latency");

Console.WriteLine("4. Average total your income");

Console.WriteLine("Please input your choice");

}

static void Main(String[] args)

{

Console.Write("Please input work days: ");

decimal wd = Convert.ToInt32(Console.ReadLine()); Console.Write("Please input leave days: ");

decimal ld = Convert.ToInt32(Console.ReadLine()); Console.Write("Please input hours per day: ");

decimal hpd = Convert.ToInt32(Console.ReadLine()); Console.Write("Please input latency hours: ");

decimal lh = Convert.ToInt32(Console.ReadLine()); Console.Write("Please input paid per hours: ");

decimal pph = Convert.ToInt32(Console.ReadLine());

bool cl = true;

mainMenu();

while (cl)

{

mainMenu();

decimal choice = Convert.ToDecimal(Console.ReadLine());

switch (choice)

{

case 1:

decimal workhours = wd \* hpd; // so gio lam trong 1 thang

Console.WriteLine("Total of work hours: " + workhours); Console.ReadLine();

break;

case 2:

decimal totalofpaid = wd \* hpd \* pph; // so tien phai tra cho tung nguoi

Console.WriteLine("Total of paid work: " + totalofpaid); Console.ReadLine();

break;

case 3:

decimal Punishmentforlatency = lh \* pph; // so tien phat vi di tre

Console.WriteLine("Punishment of latency: " + Punishmentforlatency); Console.ReadLine();

break;

case 4:

decimal Averageincome = (wd \* hpd \* pph - lh \* pph) / (wd + ld); //trung binh tien cua moi nguoi lam viec trong 1 thang//

Console.WriteLine("Average income: " + Averageincome); Console.ReadLine();

break;

case 5:

cl = false;

Console.WriteLine(" Input valid option! do it again!");

break;

}

}

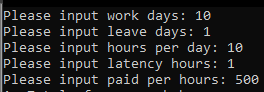
}

}

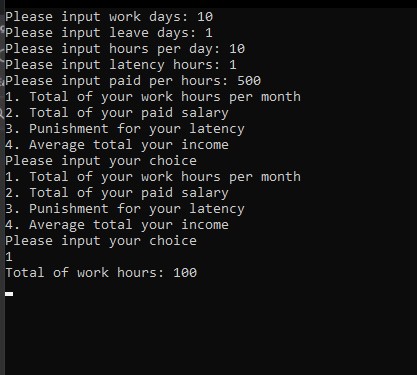
}

### Source code and screenshots of the final result

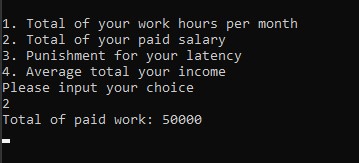
### \_Create menu, Input Values



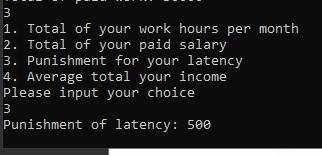
### Case 1: passed it’s working



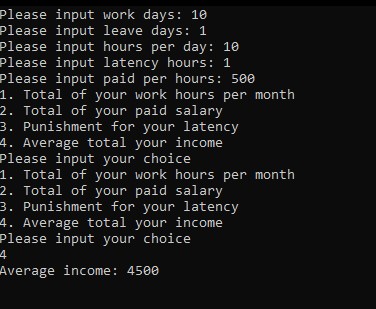
### Case 2: Also working



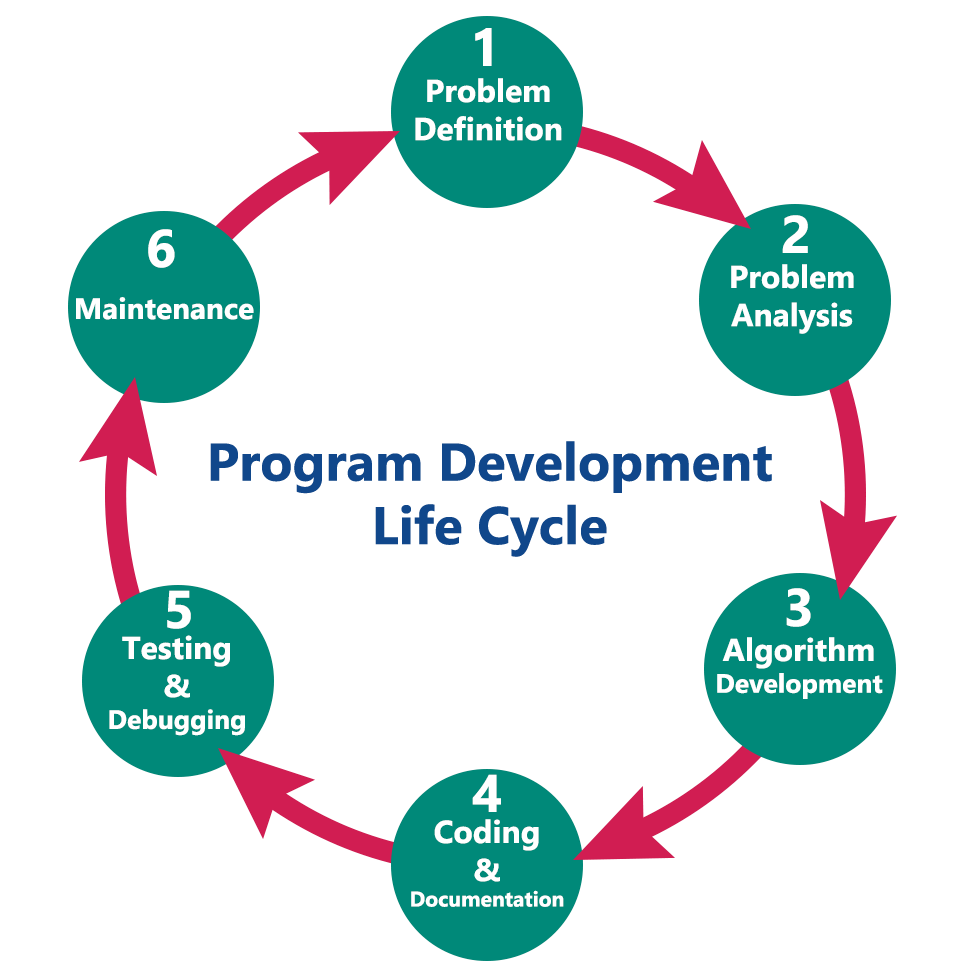
### Case 3:



### Case 4:



## 3.3 Explain briefly what is Software Development Life Cycle

(btechsmartclass.com, n.d.)

### -Steps in Program Development:

#### In general, the program development lifecycle contains 6 stages, which occur as follows.

1. Problem Definition
2. Problem Analysis
3. Algorithm Development
4. Coding & Documentation
5. Testing & Debugging
6. Maintenance

1. Define the problem

In this stage, this step, we (a programmer) study the problem, determine the problem statement and we decide the boundary of the problem. In this stage, we need to understand to determine the best way to solve these problems and report the problem of what our requirements are, the output of the problem solution. They are defined in this first phase of the program development life cycle.

Researching a problem is also necessary because it helps the programmer decide on the following:

The facts and figures needed to develop the program.

How the program will be designed

In addition, the language in which the program will be most appropriate.

What is the desired output and in what form is necessary, etc.

1. Analyze the problem

In phase 2, we define requirements such as variables, functions, etc. to solve the problem. That means we collect the resources needed to solve the problem identified during the problem identification phase. We also determine the limit of the solution.

1. Development of algorithms

In this stage, we develop a step-by-step process (Algorithm) to solve the problem using the specifications given in the previous period.

The algorithm is a series of steps that must be performed before the programmer starts preparing his program. Programmers design an algorithm to help alternative images possible in a program also.

This stage is very important for program development. That means we write solutions according to each step of the report.

The next step after designing the algorithm is to write a high-level language program. This process is called encryption.

1. Encoding & Documentation

This stage uses programming languages to write or implement practical programming instructions for the steps identified in the previous phase. In this stage, we build a reality program. That means we write a program to solve a given problem in programming languages like C, C, Java, etc.

1. Check and debug

During this stage, the program execution process for finding errors or errors is called test execution. We check to see if the code written in the previous step solved the specified problem.

Debugging is a process of detecting, locating and fixing errors in the program. It is done by running the program multiple times.

That means we check the program to see if it solves the problem for different input data values. We also check to see if it provides the desired output. It helps a programmer test the logic of the program. It also ensures that the program is error-free and workable.

1. Maintenance

When the program is completed, its document is prepared. The final document is provided to the user. It guides users on how to use the program as efficiently as possible.

In this stage, the program is actively used by users. If any improvements are found during this period, all stages will be repeated again to make improvements. That means that in this stage, the solution (program) is used by the end user. If users encounter any problems or want to have any enhancements, we need to repeat all stages from the beginning, so that the problem is solved or enhanced is added.

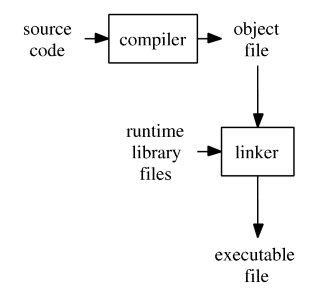
Moreover, another purpose of having documents is to allow other programmers to modify the code if necessary. Documentation should also be done in every step of the program development process

## 3.4 Explain how the source code is compiled

### 3.Explain how to source is complied

Human languages are not understood by computers. In reality, computers can only grasp numerical sequences that reflect operating codes at the most basic level (op codes for short). Humans, on the other hand, would find it extremely difficult to develop programs in terms of op codes. As a result, programming languages were created to make writing computer programs easier for humans.

Humans can read and understand programming languages. To be executed by a computer, the program (source code) must be translated into machine language (as the computer only understands machine language). The manner in which this translation takes place is determined by whether the programming language is compiled or interpreted.



# Chapter 4 - Evaluate how the problem is solved from the designed algorithm to the execution program written by a specific programming language

## 4.1 Include Test cases

### M1 Determine the steps taken from writing code to execution

-For example a assignment brief (Using code algorithms for Define the problem) about Salary Calculator Business Problem (On Company) Salary Caculator:

Input variables:

1. Work days
2. Leave days
3. Hours per day
4. Latency hours
5. Paid per hours

Input selections:

1. Total of work hours/ month
2. Total of paid/ month
3. Punishment for latency
4. Average income/ month of employer
5. Exit Programe

● Understand the problem you are trying to solve The problem:

Calculate create the result of the problem

1. Total of work hours/ month
2. Total of paid/ month
3. Punishment for latency
4. Average income/ month of employer
5. Exit Programe => It's called output

By Using:

Input variables:

1. Work days
2. Leave days
3. Hours per day
4. Latency hours
5. Paid per hours

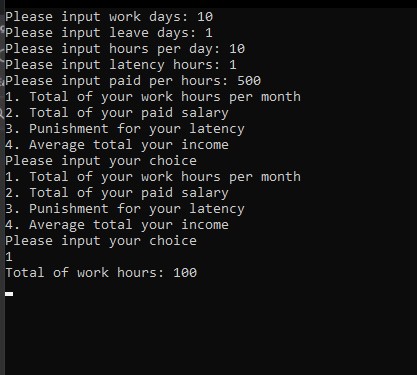
=>It's called Input

## 4.2 Evaluate how the problem is solved from the designed algorithm to the execution program written by a specific programming language

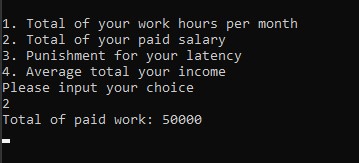
### IV. Eveluate how to problem is solved from the designed algorithm to the execution program written by a specific programming laguage

1. Test cases

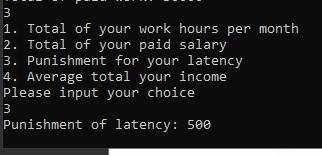
Case 1: passed it’s working



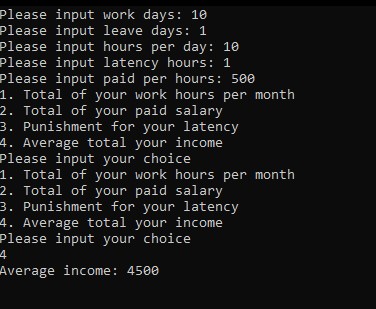
Case 2: Also working



Case 3:



Case 4:



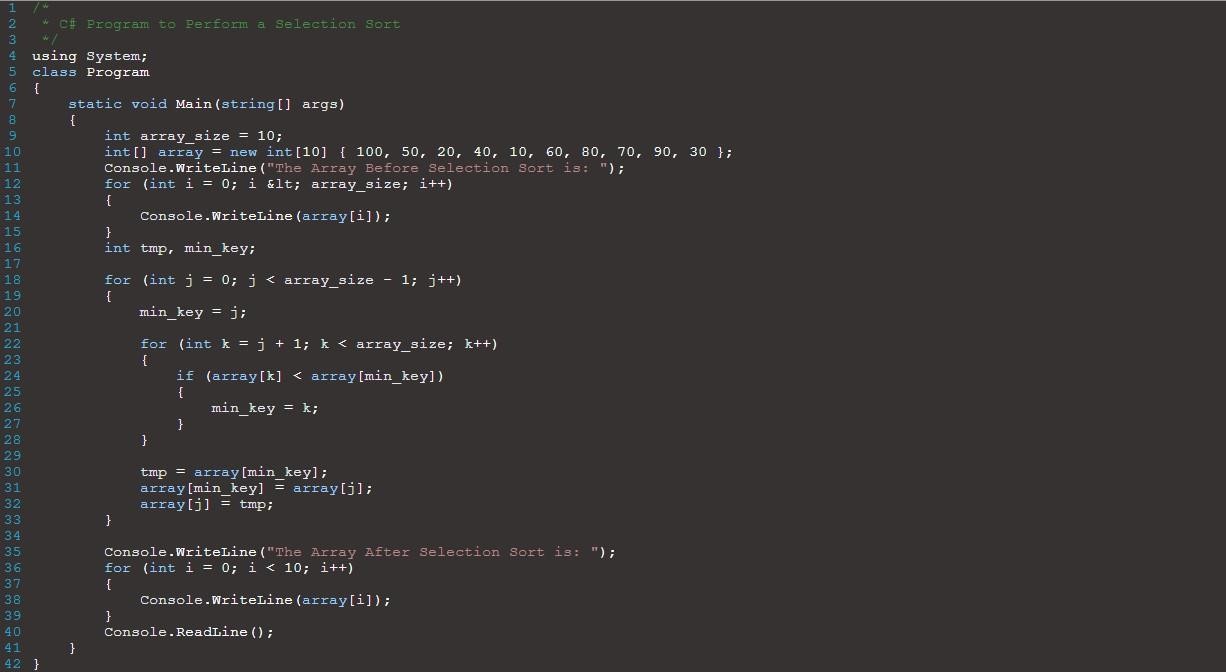
# D1 Examine the implementation of an algorithm in a suitable language. Evaluate the relationship between the written algorithm and the code variant

In several circumstances, the software has done an excellent job of solving the problem. This program, however, continues to disappoint me. I'd want to add more features, such as displaying an error message when the user inputs incorrect data types or something else, or when the user selects the incorrect option, the software will display the message "sorry, you selected the incorrect option, please try again." In the future, I'll upgrade these features in the same way.

-Examine the implementation of an algorithm in a suitable language

By C#

(/c-program-to-perform-selection-sort/, n.d.)



This is C# an algorithm program to perform Selection sort with C# code of sorting an array where it loop from the start of the loop, and check through other elements to find the minimum value. After the end of the first iteration, the minimum value is swapped with the current element.

The iteration then continues from the 2nd element and so on.

Here is the output of the C# Program:

The Array Before Selection Sort is :

100

50

20

40

10

60

80

70

90

30

The Array After Selection Sort is :

10

20

30

40

50

60

70

80

90

100

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

The code:

1. Just a normal code, with Loop, Conditional, Procedural statement
2. Code is written in a particular language.

The Algorithm

1. It's code with Certainly loop code for solve a problem faster
2. Algorithms are language independent.

The Relation

1. They are have code in their coding

# REFERENCES

## (https://www.geeksforgeeks.org/linear-search/, n.d.)