Assignment 1

Overview

You receive 6 different tasks. Before putting your hands on code it is important to analyze the objective of each task. The assignment consists on using different strategies to fully understand the scope of the task and to come up with a solution.

Exercises

1. Calculate the monthly income of an employee.
2. Calculate if a student approves or fails a course.
3. Multiplication of two values, but if any value is 0 send a message that the result is 0.
4. Division but if second value is 0 then send message that you can't divide by 0 or the result is infinite.
5. Compare two values and return the biggest value.
6. Determine if a number is odd or even.

Instructions

For each one of the exercises, perform the next tasks:

Task 1 Create a list describing the solution step by step

By using a list, describe each step a program would need to perform to solve the exercise. Keep in mind the goal is that it has to be done by a computer program, not a human being clicking stuff.

You will use this list to perform task 2.

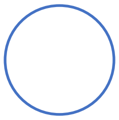
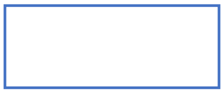
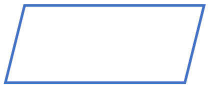
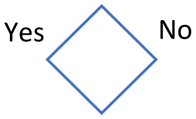
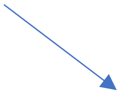
Example: Microwave a slice of pizza

01 Grab a cold slice of pizza
02 Walk to a microwave
03 Open the microwave's door
04 Put the slice inside the microwave
05 Close the microwave's door
06 Set the timer to any amount of seconds
07 Press the start button
08 Wait until the timer reaches 0
09 Open the microwave's door
10 Grab the slice of pizza carefully because it's hot
11 Close the microwave's door
12 Eat the pizza carefully because it's hot

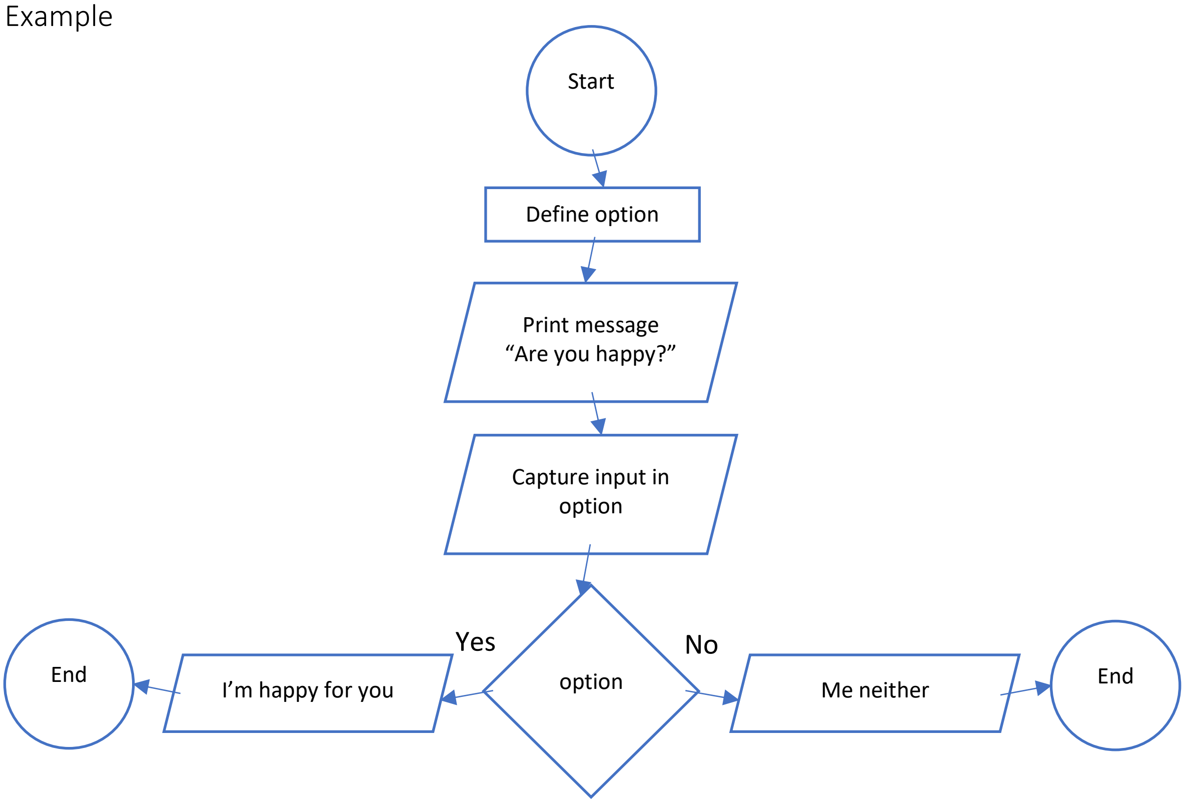
Task 2 Create the flow diagram of your solution

Flowcharts help to visualize and explain an algorithm to anyone, not just programmers.

By using the next blocks of content and the list from task 1 design the logic of your solution:

Symbol	Definition
	Defines the beginning and the ending of a process.
	Defines a new value or an operation.
	Defines input or output of data. Input means interaction with a user, output means showing something to the user.
	Defines a condition to test, if the condition is true then it continues one way, otherwise it follows the other way. You should always add something for both possibilities.
	Connects the elements and shows the path to follow.

Example



Task 3 Create the pseudocode

By using your flowchart, create a structure similar to what real code look like, but keep using words that any person can understand. Pseudocode should be transferable to any programming language.

Example

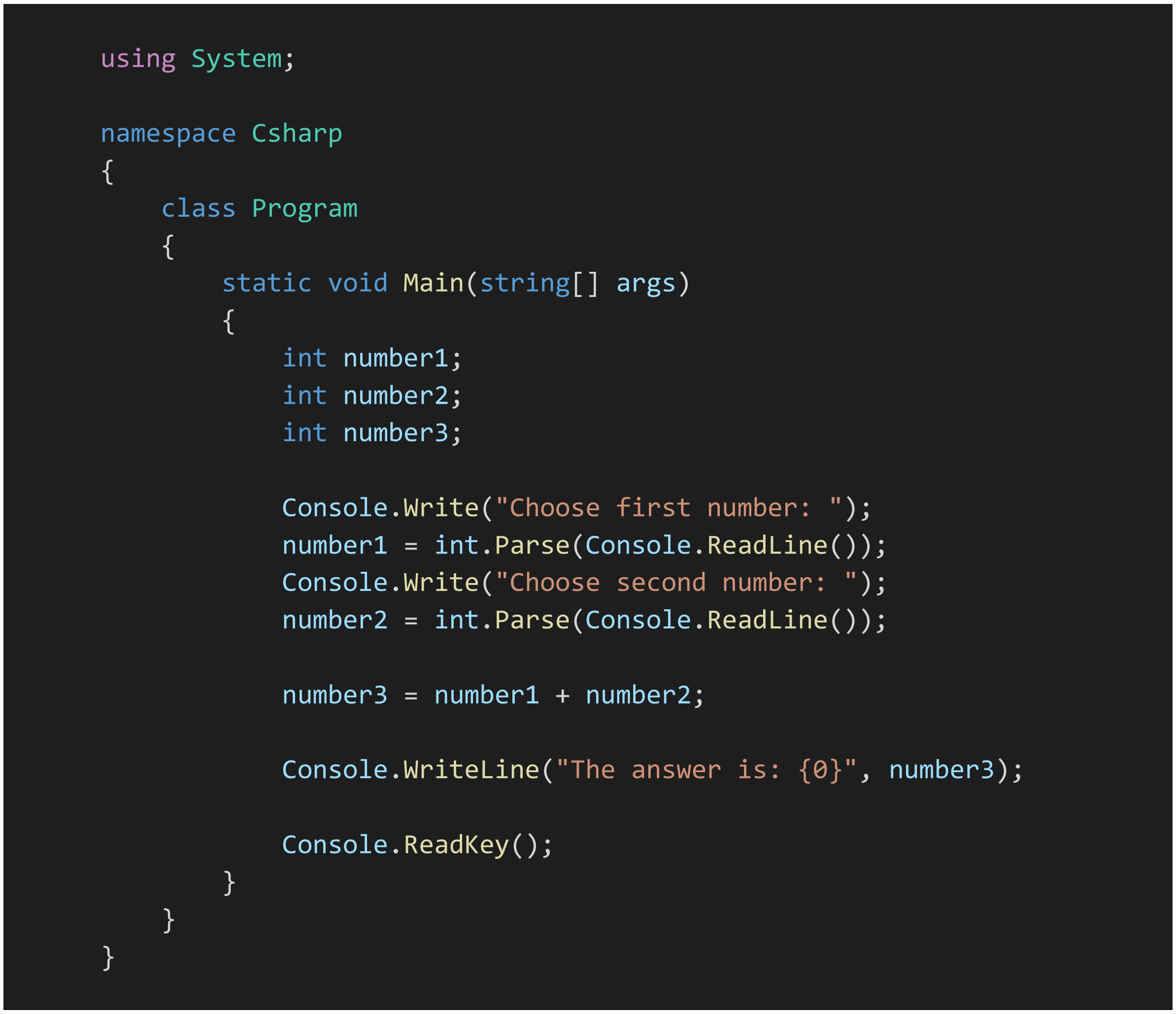
Addition of two numbers

```
START
    DEFINE NUMBER VARIABLE Number1
    DEFINE NUMBER VARIABLE Number2
    DEFINE NUMBER VARIABLE Number3

    PRINT “Choose first number”
    READ input and store in Number1
    PRINT “Choose second number”
    READ input and store in Number2

    Number3 = Number1 + Number2

    PRINT “The answer is:” Number3
END
```

Task 4 Code

Finally use the pseudocode to create your program in C#.

Example



```
using System;

namespace Csharp
{
    class Program
    {
        static void Main(string[] args)
        {
            int number1;
            int number2;
            int number3;

            Console.Write("Choose first number: ");
            number1 = int.Parse(Console.ReadLine());
            Console.Write("Choose second number: ");
            number2 = int.Parse(Console.ReadLine());

            number3 = number1 + number2;

            Console.WriteLine("The answer is: {0}", number3);

            Console.ReadKey();
        }
    }
}
```

Final result

Create a pdf document containing the 6 exercises solved using the four previous tasks.

The structure should contain at least something like:

- TITLE (Programming in C# - Assignment 1)
- Name
- Exercice 1
    - List
    - Flowchart
    - Pseudocode
    - Code in C#
- Exercice 2
    - List
    - Flowchart
    - Pseudocode
    - Code in C#
- Exercice 3
    - List
    - Flowchart
    - Pseudocode
    - Code in C#
- Exercice 4
    - List
    - Flowchart
    - Pseudocode
    - Code in C#
- Exercice 5
    - List
    - Flowchart
    - Pseudocode
    - Code in C#
- Exercice 6
    - List
    - Flowchart
    - Pseudocode
    - Code in C#

Final notes

The pdf document must be in your GitHub repository.

Due date is Monday.

RUBRIC

Definition	Value	Description
Task 1	4pts	The list describe all the steps needed to get to the result without missing anything.
Task 2	6pts	The flowchart follows a clear path and it has a start and an end.
Task 3	4pts	The pseudocode can be readable by anyone and be understood.
Task 4	6pts	The code follows the exact same structure than the pseudocode and it works. (Some elements can differ because of C#'s syntax, that's okay).
Total	20pts	