ASSIGNMENT 2
PROGRAMMING REPORT

Student performance: TRAN QUANG HUY

ID: GCD18457

Class: GCD0821

Teacher: HOANG HUU DUC

ASSIGNMENT 2 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	TRAN QUANG HUY	**Student ID**	GCD18457
Class	GCD0821	**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	HUY

Grading grid

P2	P3	P4	P5	M2	M3	M4	D2	D3	D4

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

Contents

TABLE OF FIGURES

TABLE OF TABLES

TABLE OF PICTURES

Part 1: Development Document

I. Introduction:

Patient Management Software is a general software that developed for hospital or personal clinic.

II. Scope:

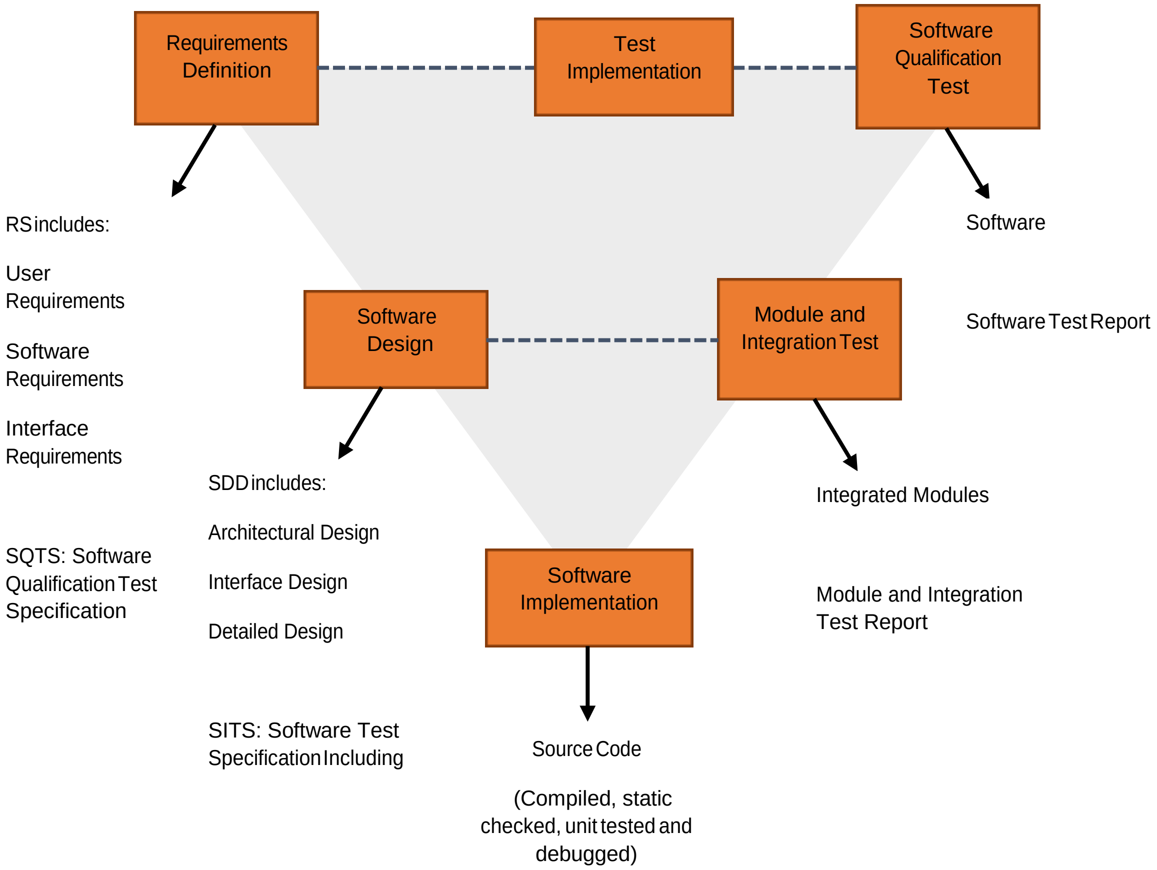


Figure 1 - Scope of program (c4learn, n.d.)

III. Overview:

Patient Management software is an application which is used to store some basic information of a patient: First name, last name, picture, address, telephone, sex, date of birth and patient record. An example: TRAN QUANG HUY, 145a Phan van truong, 0123456789, Male, 16/12/1995. And this program also provides some functions such as: add the new patient, update patient information or delete a patient record, searching patient information.

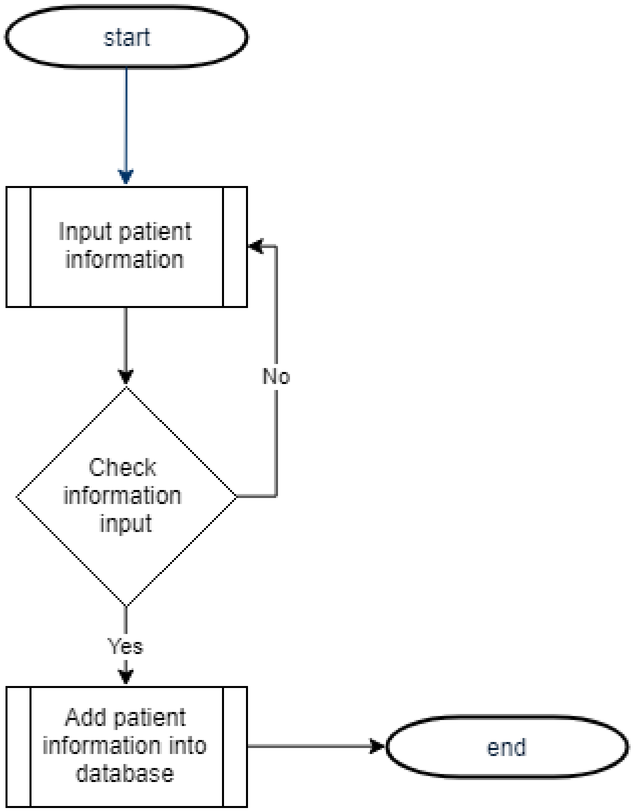
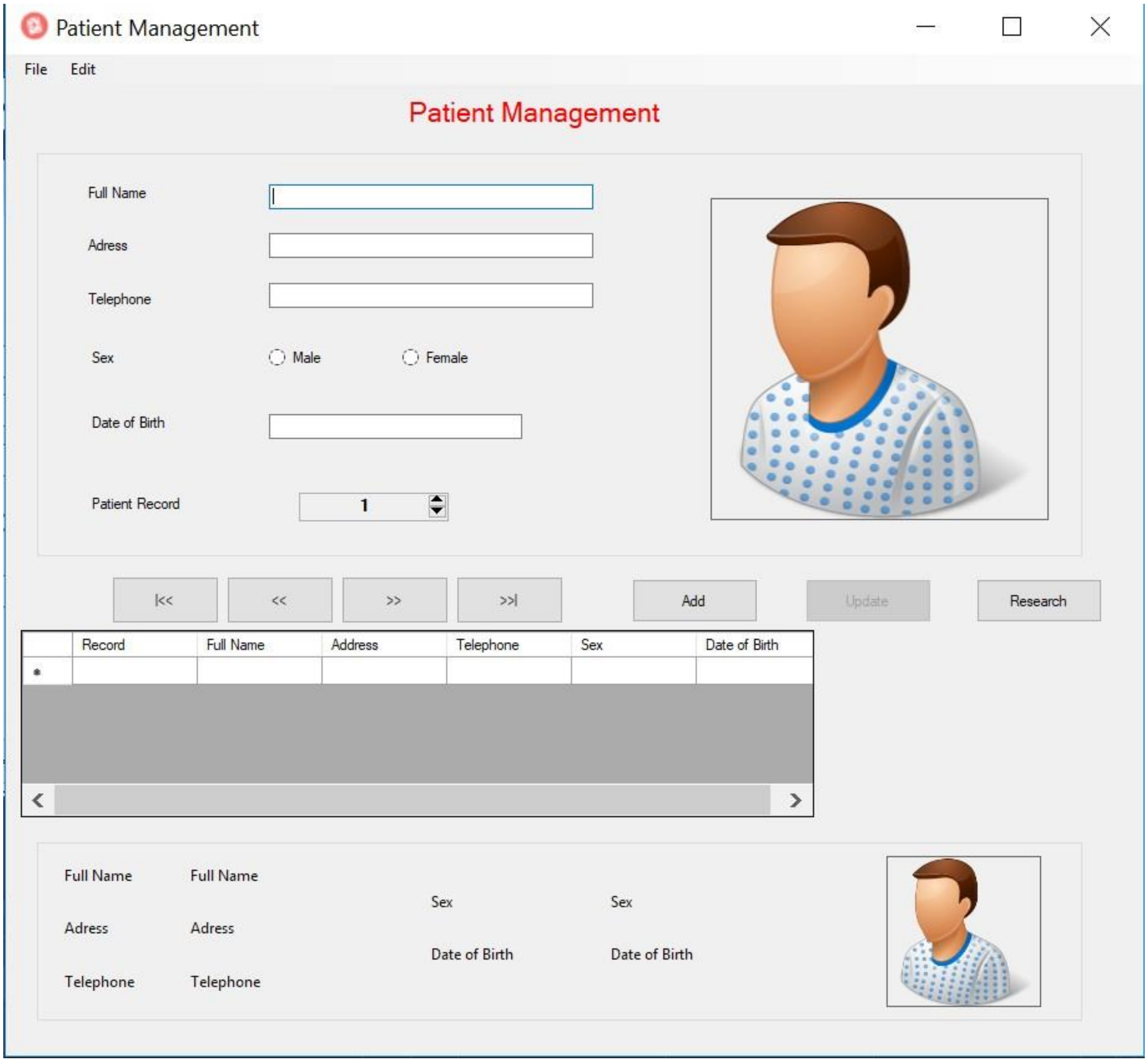

start
Input patient information
No
Check information input
Yes
Add patient information into database
end

Figure 2 - Flowchart of program

Firstly, user need to input patient information. After that, the program will check the information input is correct or not. If the input is wrong, user need to input patient again. If everything input is right, the program will add the information into database.



	Record	Full Name	Address	Telephone	Sex	Date of Birth
*						

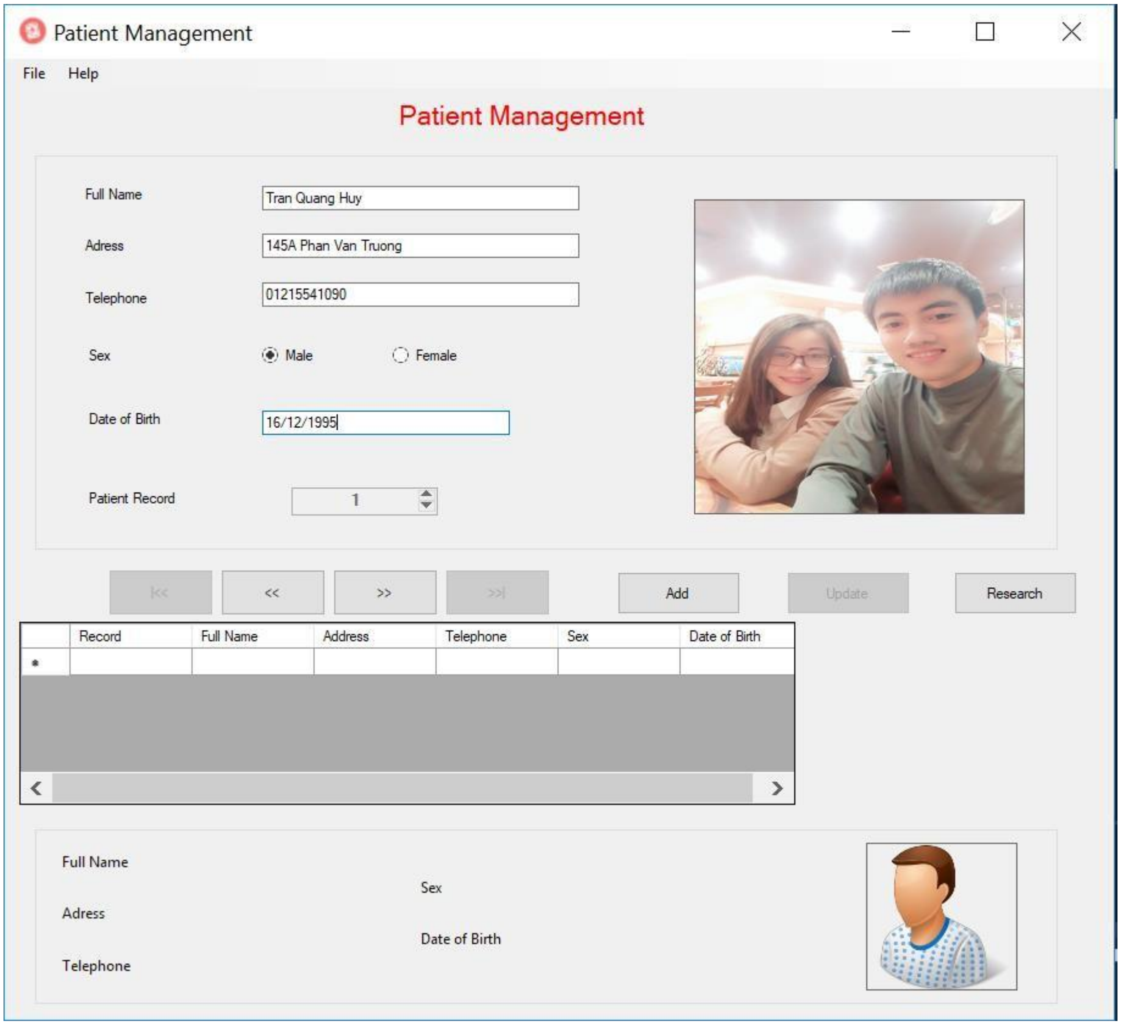
Picture 1 - The interface of Patient management

IV. User Manual:

This main window includes:

- Full Name: Require user input Patient Full Name (letters and numbers).
- Address: Require user input Patient Address (letters and numbers).
- Telephone: Require user input Patient Telephone (Letters and numbers).
- Sex: User tick one in circle Male or Female.
- Date of Birth: Require user input Patient Full Name (numbers and symbols).
- Patient Record: The Application automatic fill in “1” and increase by 1 each time.
- Picture Box: User can input Patient picture or not.

At first, user can’t click button “Update”, “First Patient” or “Last Patient” and need to fill patient information in all box:



Patient Management

Full Name: Tran Quang Huy

Adress: 145A Phan Van Truong

Telephone: 01215541090

Sex: ● Male ○ Female

Date of Birth: 16/12/1995

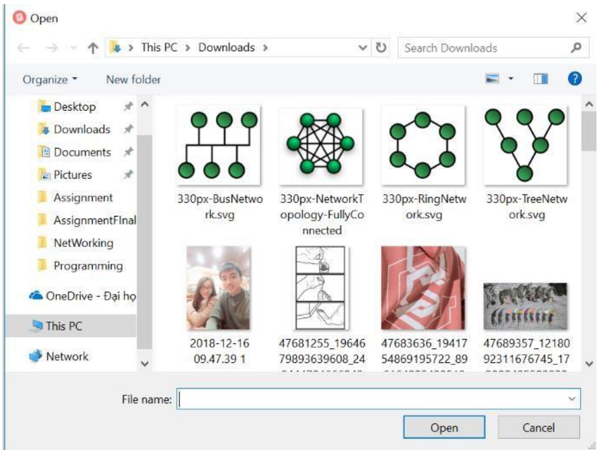
Patient Record: 1

	Record	Full Name	Address	Telephone	Sex	Date of Birth
*						

Picture 2 - Add information into box in program

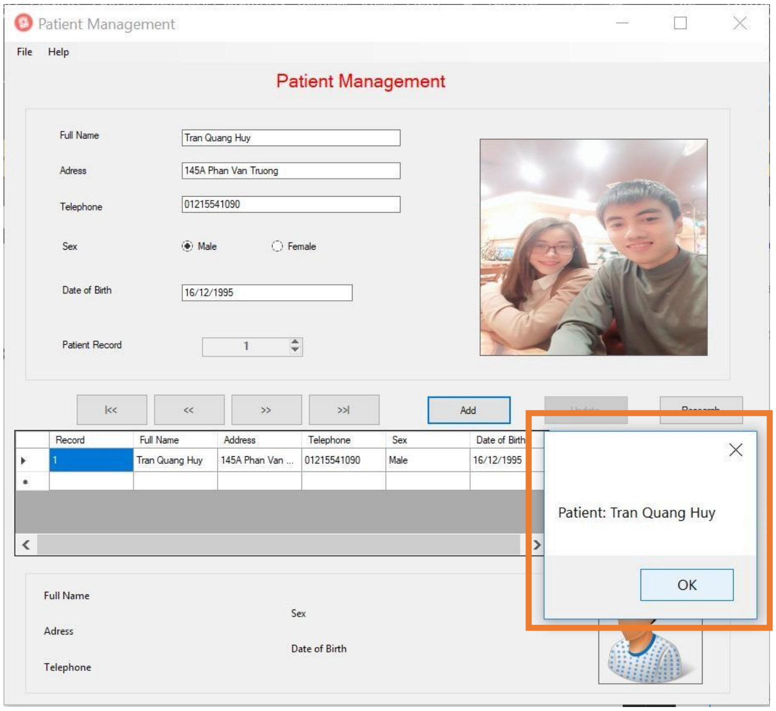
1. Button Add

- When user click Add, all information that user fill in form that recorded by application.
- If user want to add patient picture, user can click on the box image to Open picture into application, then choose the picture and click Open



Picture 3 - Select picture from computer

- The inform show up to tell user if that Information input was done.
- Click “OK” to continue.
- After that, the form will renew with Patient Record increase 1.
- When the application receives the information about Patient the button “First Patient (|<<)” and “Last Patient (>>|)” show up.



Patient Management

Full Name: Tran Quang Huy

Adress: 145A Phan Van Truong

Telephone: 01215541090

Sex: ● Male ○ Female

Date of Birth: 16/12/1995

Patient Record: 1

	Record	Full Name	Address	Telephone	Sex	Date of Birth
▸	1	Tran Quang Huy	145A Phan Van ...	01215541090	Male	16/12/1995
*						

Full Name

Adress

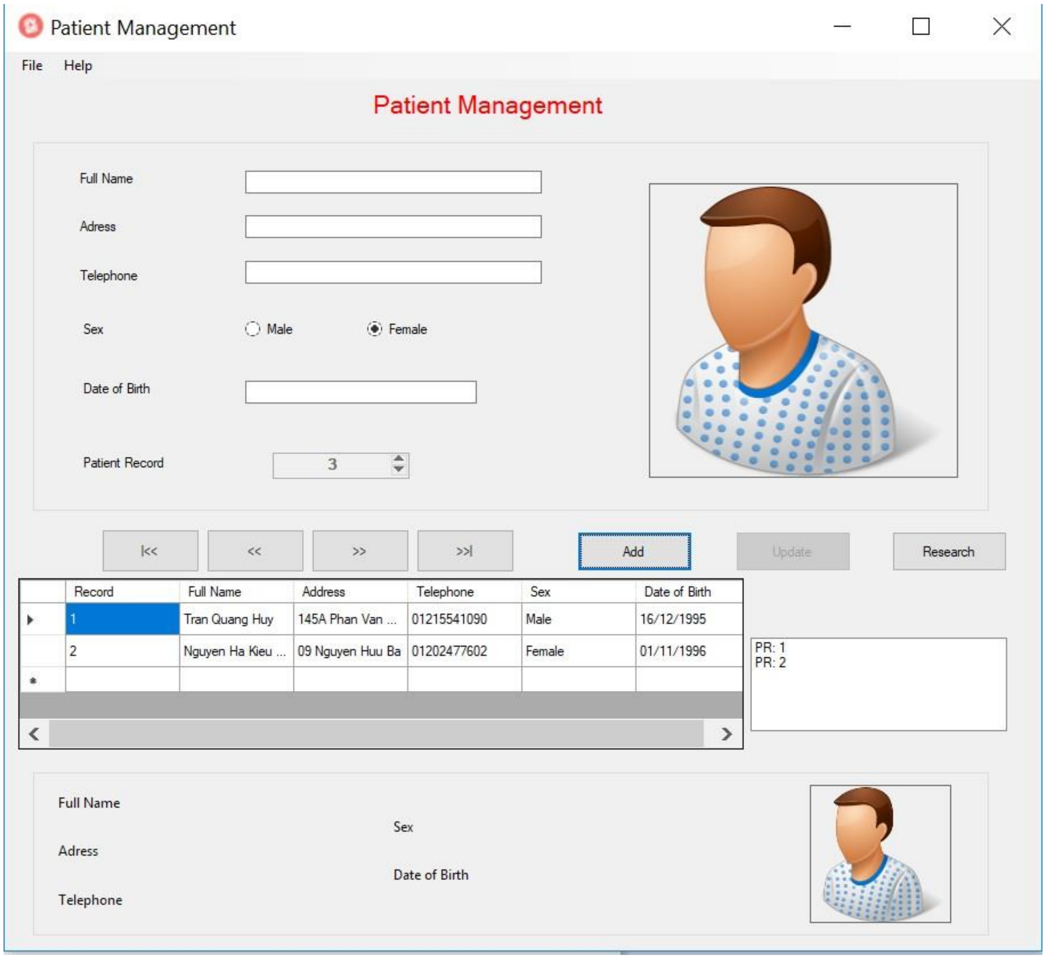
Telephone

Sex

Date of Birth

Patient: Tran Quang Huy

Picture 4 - Get notice when press Add



Full Name

Adress

Telephone

Sex ○ Male ● Female

Date of Birth

Patient Record 3

	Record	Full Name	Address	Telephone	Sex	Date of Birth
▸	1	Tran Quang Huy	145A Phan Van ...	01215541090	Male	16/12/1995
	2	Nguyen Ha Kieu ...	09 Nguyen Huu Ba	01202477602	Female	01/11/1996
*						

PR: 1
PR: 2

Full Name

Adress

Telephone

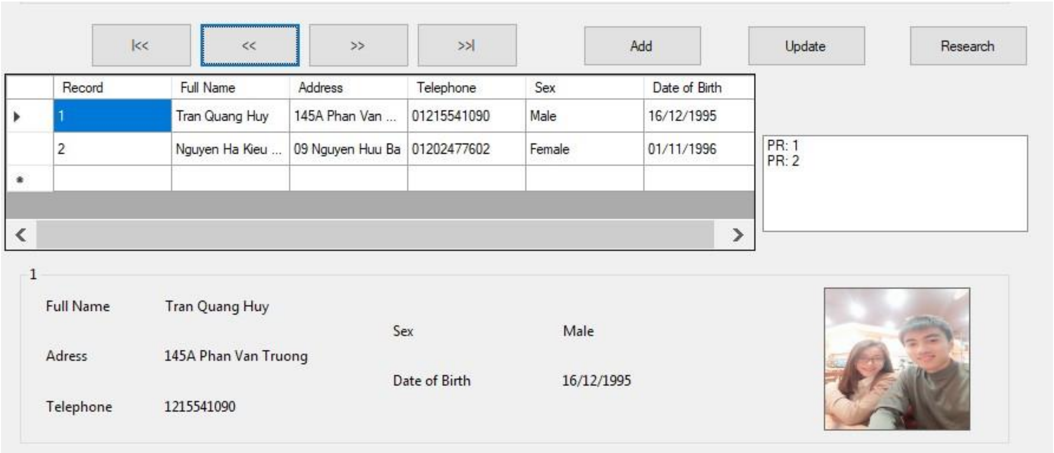
Sex

Date of Birth

Picture 5 - Patient's information in program

2. **Button “First Patient (|<<)” , “Last Patient (>>|)”, Next (>>), previous (<<).**

- User can click the Order Button now to show, next, previous, Last or First Patient
- The information will show up in bottom box.



	Record	Full Name	Address	Telephone	Sex	Date of Birth
▸	1	Tran Quang Huy	145A Phan Van ...	01215541090	Male	16/12/1995
	2	Nguyen Ha Kieu ...	09 Nguyen Huu Ba	01202477602	Female	01/11/1996
*						

PR: 1
PR: 2

1

Full Name Tran Quang Huy

Adress 145A Phan Van Truong

Telephone 1215541090

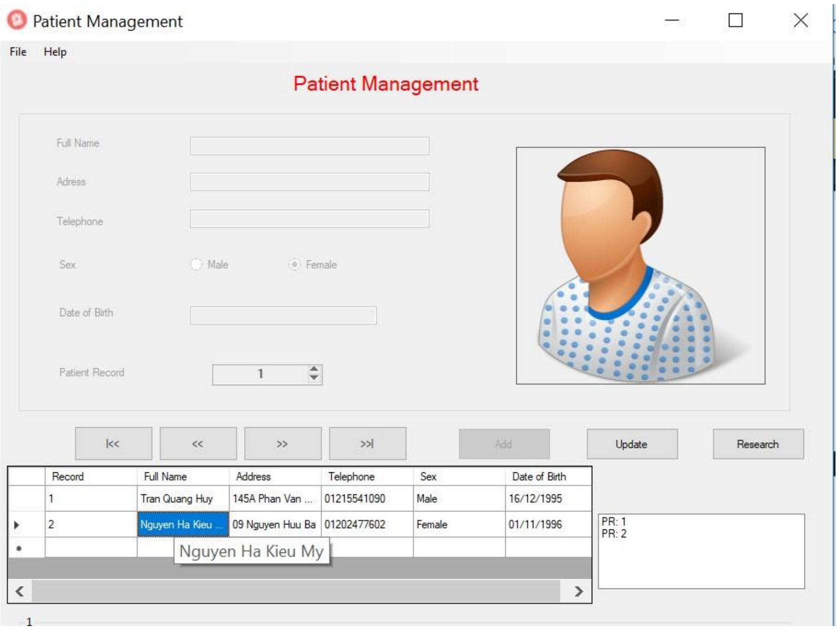
Sex Male

Date of Birth 16/12/1995

Picture 6 - Button First patient, last patient, next and previous

3. Edit information that user fill in.

- When user click in any Patient Information that will make “Box Fill in information” and “button Add” disable.



Patient Management

Full Name

Adress

Telephone

Sex ○ Male ● Female

Date of Birth

Patient Record 1

	Record	Full Name	Address	Telephone	Sex	Date of Birth
	1	Tran Quang Huy	145A Phan Van ...	01215541090	Male	16/12/1995
▸	2	Nguyen Ha Kieu ...	09 Nguyen Huu Ba	01202477602	Female	01/11/1996
*						

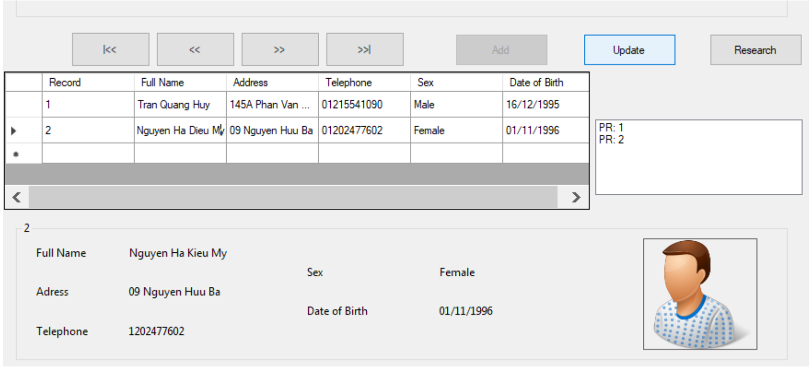
Nguyen Ha Kieu My

PR: 1
PR: 2

1

Picture 7 - Edit patient's information

- Double click in information to edit.



	Record	Full Name	Address	Telephone	Sex	Date of Birth
	1	Tran Quang Huy	145A Phan Van ...	01215541090	Male	16/12/1995
▸	2	Nguyen Ha Dieu My	09 Nguyen Huu Ba	01202477602	Female	01/11/1996
*						

PR: 1
PR: 2

2

Full Name Nguyen Ha Kieu My

Adress 09 Nguyen Huu Ba

Telephone 1202477602

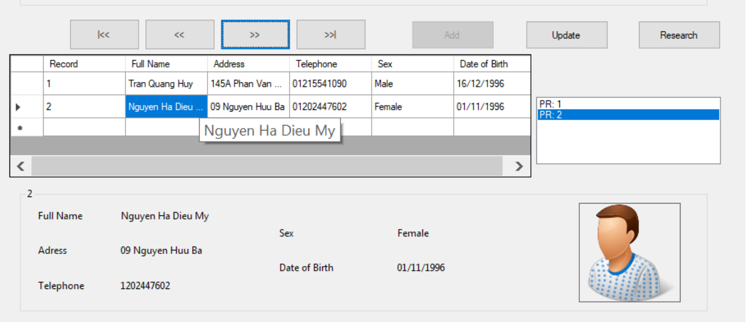
Sex Female

Date of Birth 01/11/1996

Picture 8 - Edit patient's information

4. **Update Patient information.**

- After finish edit information then click “button Update” to tell application know that you have done, and the application will record new Patient information
- If user want to comeback add more patient information: press key: “Esc” on keyboard.



	Record	Full Name	Address	Telephone	Sex	Date of Birth
	1	Tran Quang Huy	145A Phan Van ...	01215541090	Male	16/12/1996
▸	2	Nguyen Ha Dieu ...	09 Nguyen Huu Ba	01202447602	Female	01/11/1996
*						

Nguyen Ha Dieu My

PR: 1
PR: 2

2

Full Name Nguyen Ha Dieu My

Sex Female

Adress 09 Nguyen Huu Ba

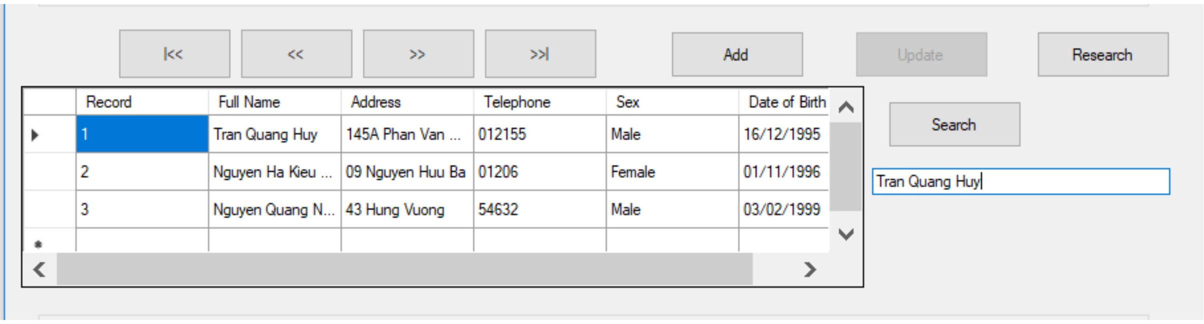
Date of Birth 01/11/1996

Telephone 1202447602

Picture 9 - Update patient's information

5. Search Patient Information (Name only).

- Click "button Research" for searching Patient.
- Fill in new box appear about Name of patient that user want to find.

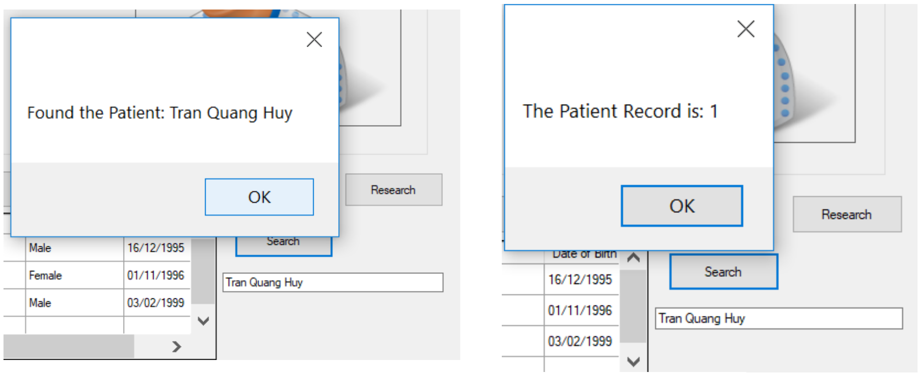


Record	Full Name	Address	Telephone	Sex	Date of Birth
1	Tran Quang Huy	145A Phan Van ...	012155	Male	16/12/1995
2	Nguyen Ha Kieu ...	09 Nguyen Huu Ba	01206	Female	01/11/1996
3	Nguyen Quang N...	43 Hung Vuong	54632	Male	03/02/1999

Tran Quang Huy

Picture 10 - Search patient's information

- Application find the patient that user want to find and inform to user know that patient record, if there is no one have the same name the application will inform that can't find that patient



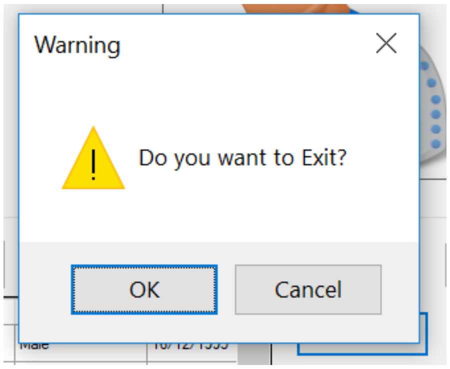
Found the Patient: Tran Quang Huy

The Patient Record is: 1

Picture 11 - Get notice about Search function

6. The menu strips.

- Menu strip show to user to:
  - New input (Ctrl +N): To renew the form fill in and increase the record 1.
  - Exit (Alt +F4): To exit this application**.**



Warning

Do you want to Exit?

Picture 12 - Menu strips

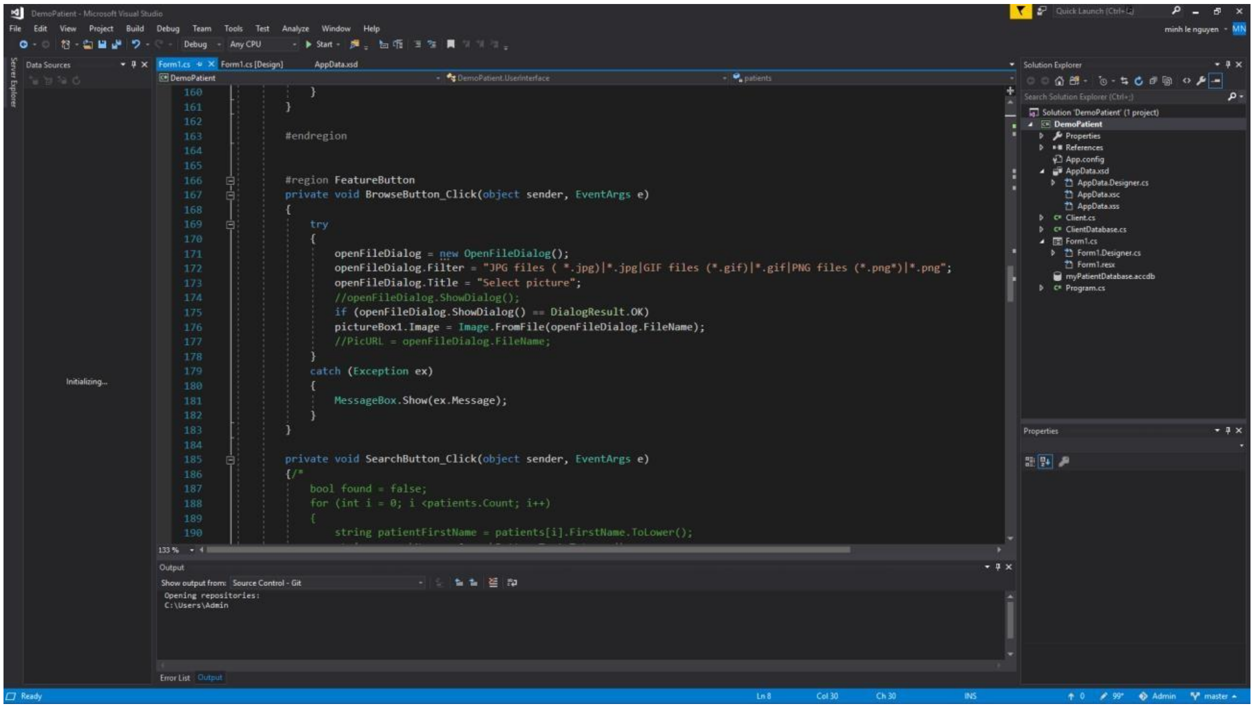
Part 2: IDE evaluation

I. Introduction

IDE as known as Integrated Development Environment which is a software application provides a toolbox that include all the facilities required to computer programmers for software development. I find IDE make it easier for a programmer to build an application using its comprehension facilities. (wikipedia, n.d.)

An IDE usually contains source editor, debugger, build automation. Most of modern IDE has intelligent code completion. This feature help speeds-up the process of coding by minimizing typos and other common mistakes, the process of development will be accelerated in result.

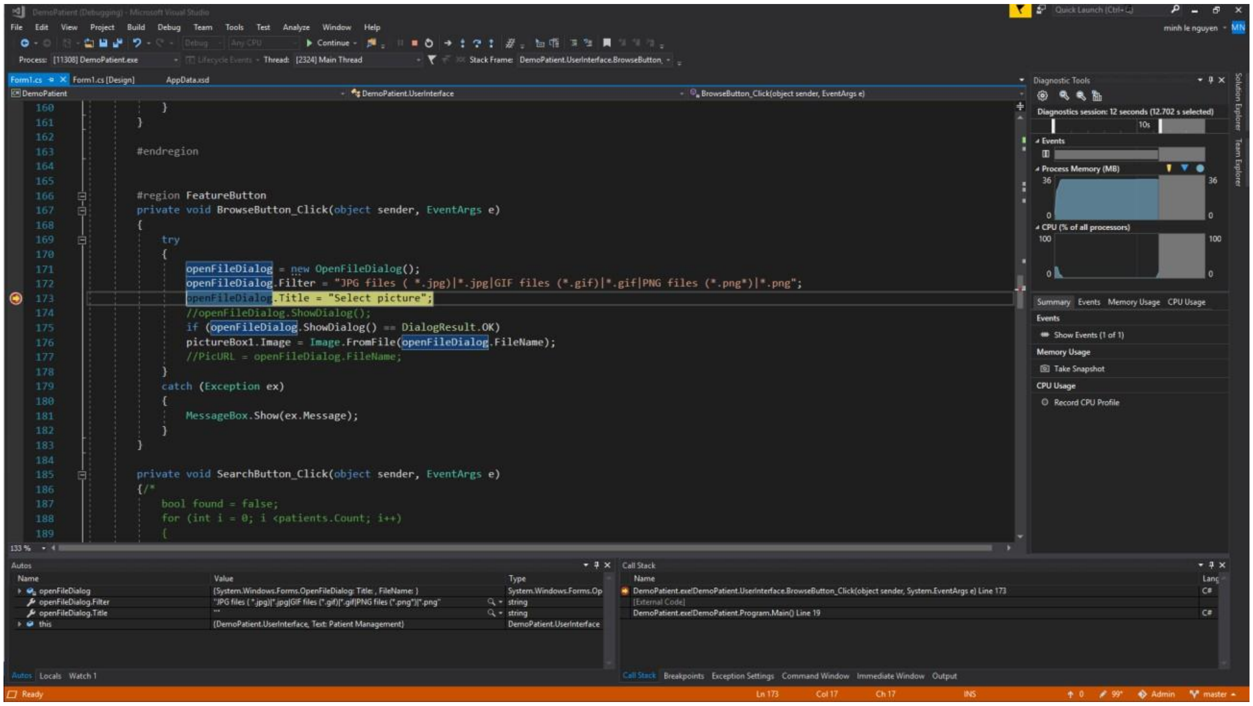
This is how Source editor and Debugger looks like in Microsoft Visual Studio 2017



```
            }
        }

        #endregion


        #region FeatureButton
        private void BrowseButton_Click(object sender, EventArgs e)
        {
            try
            {
                openFileDialog = new OpenFileDialog();
                openFileDialog.Filter = "JPG files ( *.jpg)|*.jpg|GIF files (*.gif)|*.gif|PNG files (*.png*)|*.png";
                openFileDialog.Title = "Select picture";
                //openFileDialog.ShowDialog();
                if (openFileDialog.ShowDialog() == DialogResult.OK)
                pictureBox1.Image = Image.FromFile(openFileDialog.FileName);
                //PicURL = openFileDialog.FileName;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        private void SearchButton_Click(object sender, EventArgs e)
        {/*
            bool found = false;
            for (int i = 0; i <patients.Count; i++)
            {
                string patientFirstName = patients[i].FirstName.ToLower();
```

```
Opening repositories:
C:\Users\Admin
```

Picture 13 - Visual studio interface



DemoPatient (Debugging) - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Continue
Process: [11308] DemoPatient.exe
Thread: [2324] Main Thread
Stack Frame: DemoPatient.UserInterface.BrowseButton_
Form1.cs
Form1.cs [Design]
AppData.xsd
DemoPatient
DemoPatient.UserInterface
BrowseButton_Click(object sender, EventArgs e)
Diagnostic Tools
Diagnostics session: 12 seconds (12.702 s selected)
Events
Process Memory (MB)
CPU (% of all processors)
Summary Events Memory Usage CPU Usage
Show Events (1 of 1)
Take Snapshot
Record CPU Profile
Autos Locals Watch 1
Call Stack Breakpoints Exception Settings Command Window Immediate Window Output
Ready
Ln 173 Col 17 Ch 17 INS
Admin master

```
            }
        }

        #endregion


        #region FeatureButton
        private void BrowseButton_Click(object sender, EventArgs e)
        {
            try
            {
                openFileDialog = new OpenFileDialog();
                openFileDialog.Filter = "JPG files ( *.jpg)|*.jpg|GIF files (*.gif)|*.gif|PNG files (*.png*)|*.png";
                openFileDialog.Title = "Select picture";
                //openFileDialog.ShowDialog();
                if (openFileDialog.ShowDialog() == DialogResult.OK)
                pictureBox1.Image = Image.FromFile(openFileDialog.FileName);
                //PicURL = openFileDialog.FileName;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        private void SearchButton_Click(object sender, EventArgs e)
        {/*
            bool found = false;
            for (int i = 0; i <patients.Count; i++)
            {
```

Autos

Name	Value	Type
openFileDialog	{System.Windows.Forms.OpenFileDialog: Title: , FileName: }	System.Windows.Forms.Op
openFileDialog.Filter	"JPG files (*.jpg)|*.jpg|GIF files (*.gif)|*.gif|PNG files (*.png*)|*.png"	string
openFileDialog.Title	""	string
this	{DemoPatient.UserInterface, Text: Patient Management}	DemoPatient.UserInterface

Call Stack

Name	Lang
DemoPatient.exe!DemoPatient.UserInterface.BrowseButton_Click(object sender, System.EventArgs e) Line 173	C#
[External Code]	
DemoPatient.exe!DemoPatient.Program.Main() Line 19	C#

Picture 14 - Run program in visual studio

II. Implement basic algorithm in code using an IDE:

For example, this is a program finds the largest number in a set of number which in C# program language using Visual Studio 2017

*** Algorithm:**

- Check if there are no number in the set so there is no largest number.

- Assume the first number in the set numbers is the largest.

- Compare the current largest number for next number in the set. If this number is larger than current highest number, then set this number to be the largest number.

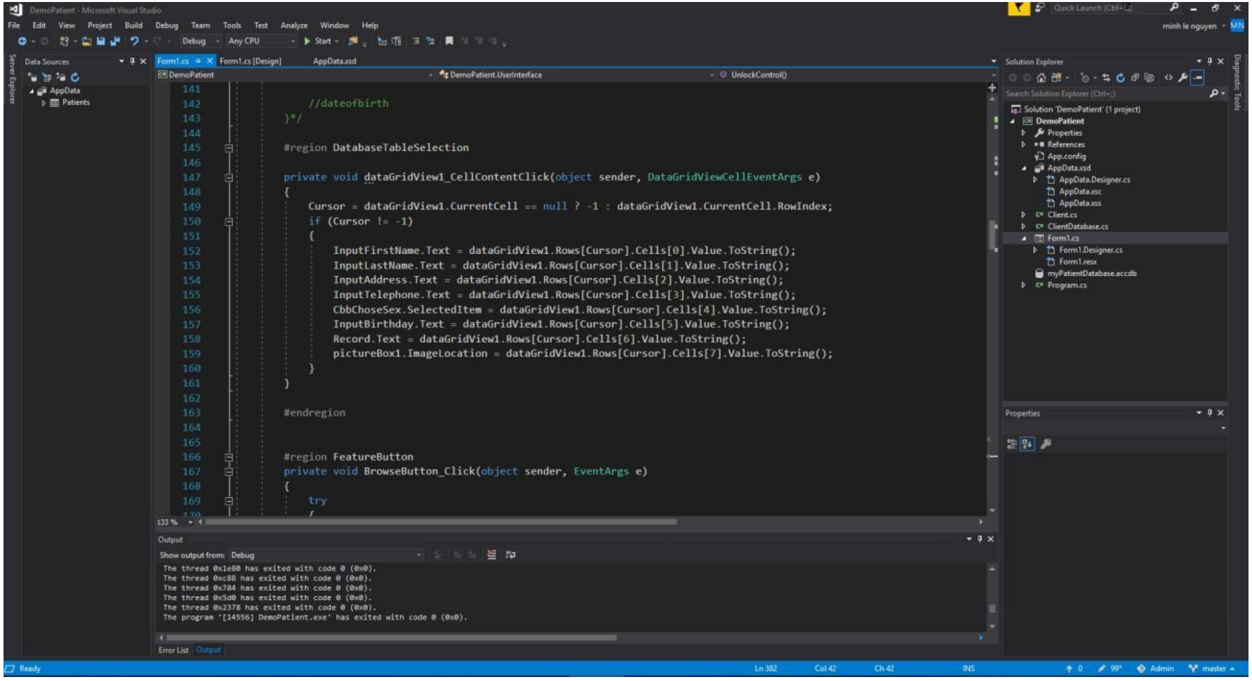
- Keep doing step three until there is no number in the set, we can consider that current largest number is the largest number in the set.(lynda, n.d.)

* Code in C# program language using Visual Studio 2017

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Factorial_Do_While
{
    class Program
    {
        static void Main(string[] args)
        {
            int n;
            bool Key = true;

            while (Key) ;
            Console.Write("Input your number: ");
            n = Int32.Parse(Console.ReadLine());

            if (n==0)
            {
                Key = false;
                Console.WriteLine("Good bye!!!");
            }
```

III. Using IDE and without using IDE:

- **Using IDE:**



```
        //dateofbirth
    }*/

    #region DatabaseTableSelection

    private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
    {
        Cursor = dataGridView1.CurrentCell == null ? -1 : dataGridView1.CurrentCell.RowIndex;
        if (Cursor != -1)
        {
            InputFirstName.Text = dataGridView1.Rows[Cursor].Cells[0].Value.ToString();
            InputLastName.Text = dataGridView1.Rows[Cursor].Cells[1].Value.ToString();
            InputAddress.Text = dataGridView1.Rows[Cursor].Cells[2].Value.ToString();
            InputTelephone.Text = dataGridView1.Rows[Cursor].Cells[3].Value.ToString();
            CbbChoseSex.SelectedItem = dataGridView1.Rows[Cursor].Cells[4].Value.ToString();
            InputBirthday.Text = dataGridView1.Rows[Cursor].Cells[5].Value.ToString();
            Record.Text = dataGridView1.Rows[Cursor].Cells[6].Value.ToString();
            pictureBox1.ImageLocation = dataGridView1.Rows[Cursor].Cells[7].Value.ToString();
        }
    }

    #endregion


    #region FeatureButton
    private void BrowseButton_Click(object sender, EventArgs e)
    {
        try
        {
```

```
The thread 0x1e80 has exited with code 0 (0x0).
The thread 0xc88 has exited with code 0 (0x0).
The thread 0x784 has exited with code 0 (0x0).
The thread 0x5d0 has exited with code 0 (0x0).
The thread 0x2378 has exited with code 0 (0x0).
The program '[14556] DemoPatient.exe' has exited with code 0 (0x0).
```

Picture 15 - Using visual studio to code

This is how Microsoft Visual Studio 2017 looks like

With IDE, it provides for programmer some features and tools to develop a software. It helps to increasing their productivity, reducing time to debug and finding solution for problem. In details, IDEs good for:

1. Quickly navigating to type without needing to worry about namespace, class, project, etc.
2. Auto complete when you cannot remember the names of a function or methods (With key: Tab)
3. Automatic code generation
4. Refactoring
5. Organize imports (automatically adding appropriate imports in Java, using directives in C#)
6. Warning-as-you-type (Example: some errors don't even require a compile cycle)
7. Hovering over something to see the tip
8. Keeping a view of files, errors/warnings/console/unit tests and source code all on the screen at the same time in a useful way
9. Ease of running unit tests from the same window
10. Integrated debugging
11. Integrated source control

But there are also some negative points when using IDE (in my opinion):

1. We will be dependence on them
2. IDEs are complicated tools. To maximize their benefits will require time and patience
3. Will not fix bad code, practices, or design: IDE are like paintbrushes, it will help you to fix the code design automatically, so you are very dependent on it. In short, you need to create a masterpiece by your skill and decisions

- **Without using IDE (Text Editor):**

Advantages:

1. No more distractions
2. After saving your code in the text editor and run it from command prompt. It will give you a true sense of programming and execution flow
3. Without auto-completion feature, you will be learned and taught yourself what to write on next line of the code
4. When you run the program, it may get the code syntax or logical error. Analyzing and fixing these bugs will improve you debugging skill

Disadvantage:

1. Not for beginners, if you want to write a website, it requires basic knowledge of HTML tags for a user.
2. It may take more time to add style and specific formal

Part 3: Debugging Evaluation

I. What is Debugging:

Debugging is the process that help developer locate and remove computer program bugs, errors or abnormalities that affect program's working and its result, which is handled by debugging tools. Debugging checks, detect and corrects errors or bugs allow the program works properly and effectively (ee, n.d.)

Almost of IDEs provide tools to debug your program such as Eclipse, Visual Studio, Android Studio, Net Beans etc. The steps to debugging commonly depend on the type of error that you are trying to debug. I will clarify two types of debugging process. They are bit similar:

- Run-time error.

- Logical error.

1. Run-Time Error:

For example, if you are stuck in *DevideByZeroException*, then you can add a breakpoint to this function and run it in debug mode, you can see it execute step by step and look into how your code gets a variable with zero. This type is Run-time Error. *DevideByZeroException* are found at run-time when the program executed.

2. Logical Error Debugging:

Logical error somehow is a tough task because they cannot be found easily. A logical error is a bug that make the program run in correctly, but not to execute abnormally (or crash like Run-time error). This error produces unexpected error or other behavior, and you cannot recognize it immediately. For example, you write a C# function that calculate the sum of two numbers.

```
int Sum(int A, int B)
{
    return A + B;
  }
```

This code contains a logic error. It is missing parentheses in the calculation, so when code execution, it doesn't give you an expected result.

3. Process:

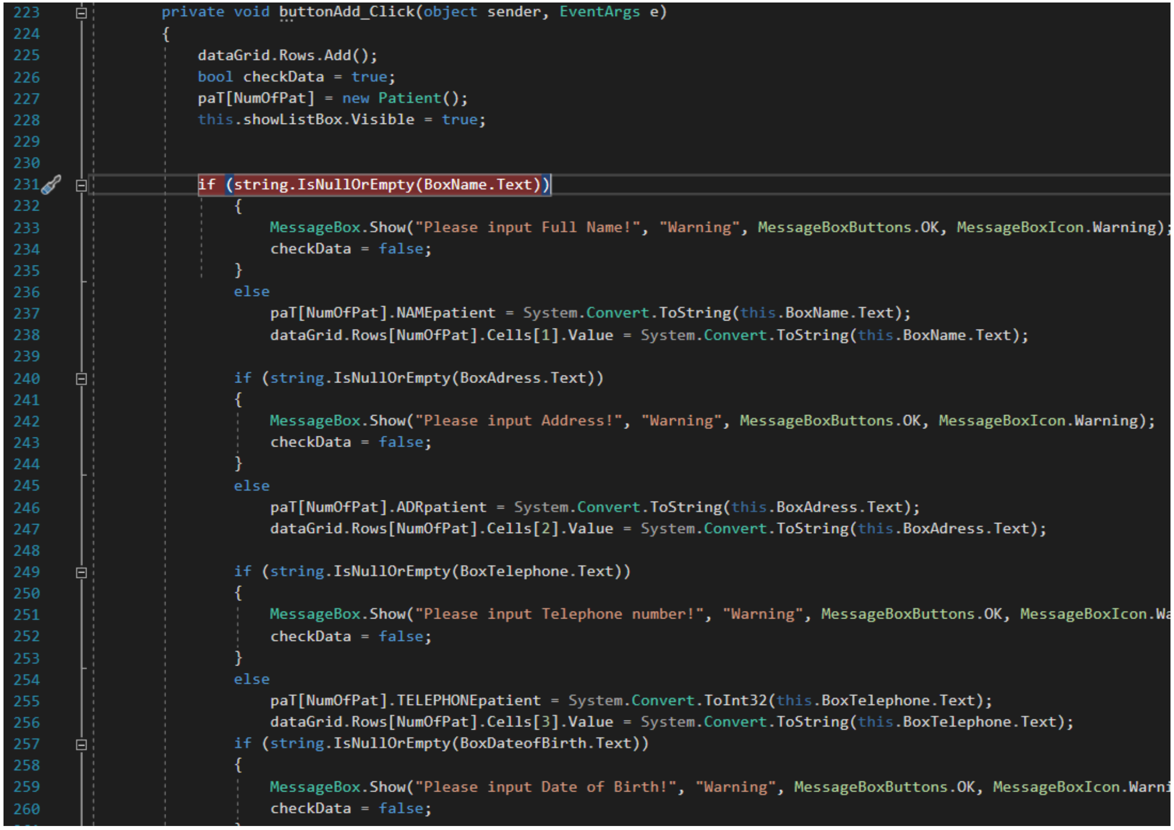
In every IDE, you can choose one or more breakpoints is your option. Breakpoints are a few steps in your code where code execution stop when it reaches that place (or point). Then you can check application's memory variables, consumption. At this time, you can check what is the status of application and how it should behave. You can also check what type of the variables and how they are being used. This would help you in fixing the problem.

4. Facilities available in the IDE:

Let me show you a specific example to know about facilities available in the IDE, Most of IDEs have the same debugging tools so I will use Microsoft Visual Studio 2017

To debug, you need to start your program and set one or more breakpoint that you want to check and start the debugger

1. In line 231, set a breakpoint by clicking the left margin of the first line or clicking on line 231 and press F9.

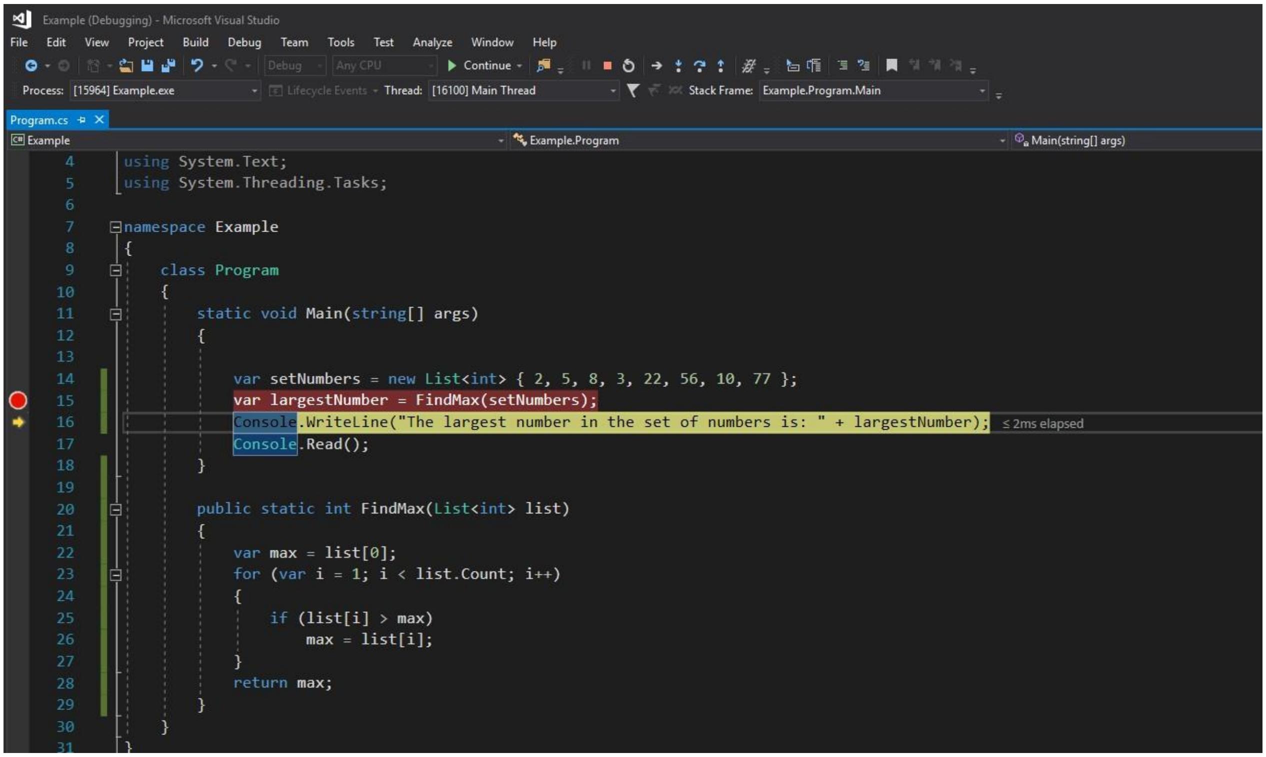


```
223         private void buttonAdd_Click(object sender, EventArgs e)
224         {
225             dataGrid.Rows.Add();
226             bool checkData = true;
227             paT[NumOfPat] = new Patient();
228             this.showListBox.Visible = true;
229 
230 
231             if (string.IsNullOrEmpty(BoxName.Text))
232                 {
233                     MessageBox.Show("Please input Full Name!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
234                     checkData = false;
235                 }
236                 else
237                     paT[NumOfPat].NAMEpatient = System.Convert.ToString(this.BoxName.Text);
238                     dataGrid.Rows[NumOfPat].Cells[1].Value = System.Convert.ToString(this.BoxName.Text);
239 
240                 if (string.IsNullOrEmpty(BoxAdress.Text))
241                 {
242                     MessageBox.Show("Please input Address!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
243                     checkData = false;
244                 }
245                 else
246                     paT[NumOfPat].ADRpatient = System.Convert.ToString(this.BoxAdress.Text);
247                     dataGrid.Rows[NumOfPat].Cells[2].Value = System.Convert.ToString(this.BoxAdress.Text);
248 
249                 if (string.IsNullOrEmpty(BoxTelephone.Text))
250                 {
251                     MessageBox.Show("Please input Telephone number!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Wa
252                     checkData = false;
253                 }
254                 else
255                     paT[NumOfPat].TELEPHONEpatient = System.Convert.ToInt32(this.BoxTelephone.Text);
256                     dataGrid.Rows[NumOfPat].Cells[3].Value = System.Convert.ToString(this.BoxTelephone.Text);
257                 if (string.IsNullOrEmpty(BoxDateofBirth.Text))
258                 {
259                     MessageBox.Show("Please input Date of Birth!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warni
260                     checkData = false;
```

Picture 16 - Set a breakpoint in studio

2. Press F5 (Debug/ Start Debugging) to execute the code in debug mode, and the debuggers runs to the line where you put the breakpoint

3. Press F10 to *Step Over* which means to execute the code line by line

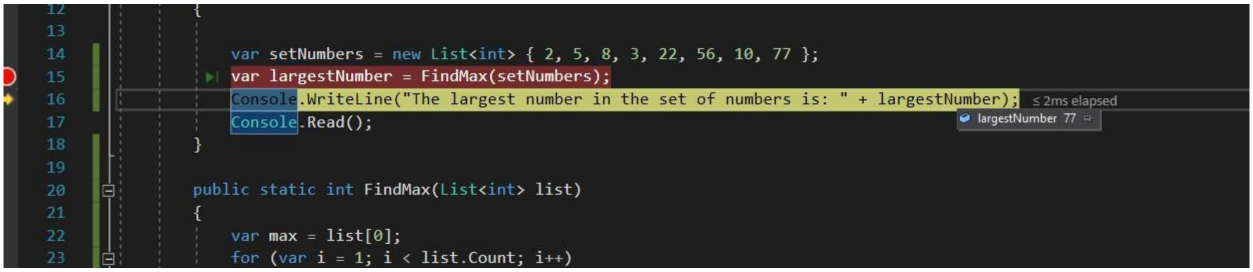


Process: [15964] Example.exe Thread: [16100] Main Thread Stack Frame: Example.Program.Main

```
using System.Text;
using System.Threading.Tasks;

namespace Example
{
    class Program
    {
        static void Main(string[] args)
        {

            var setNumbers = new List<int> { 2, 5, 8, 3, 22, 56, 10, 77 };
            var largestNumber = FindMax(setNumbers);
            Console.WriteLine("The largest number in the set of numbers is: " + largestNumber);  ≤ 2ms elapsed
            Console.Read();
        }

        public static int FindMax(List<int> list)
        {
            var max = list[0];
            for (var i = 1; i < list.Count; i++)
            {
                if (list[i] > max)
                    max = list[i];
            }
            return max;
        }
    }
}
```

Picture 17 - Step Over in Visual Studio

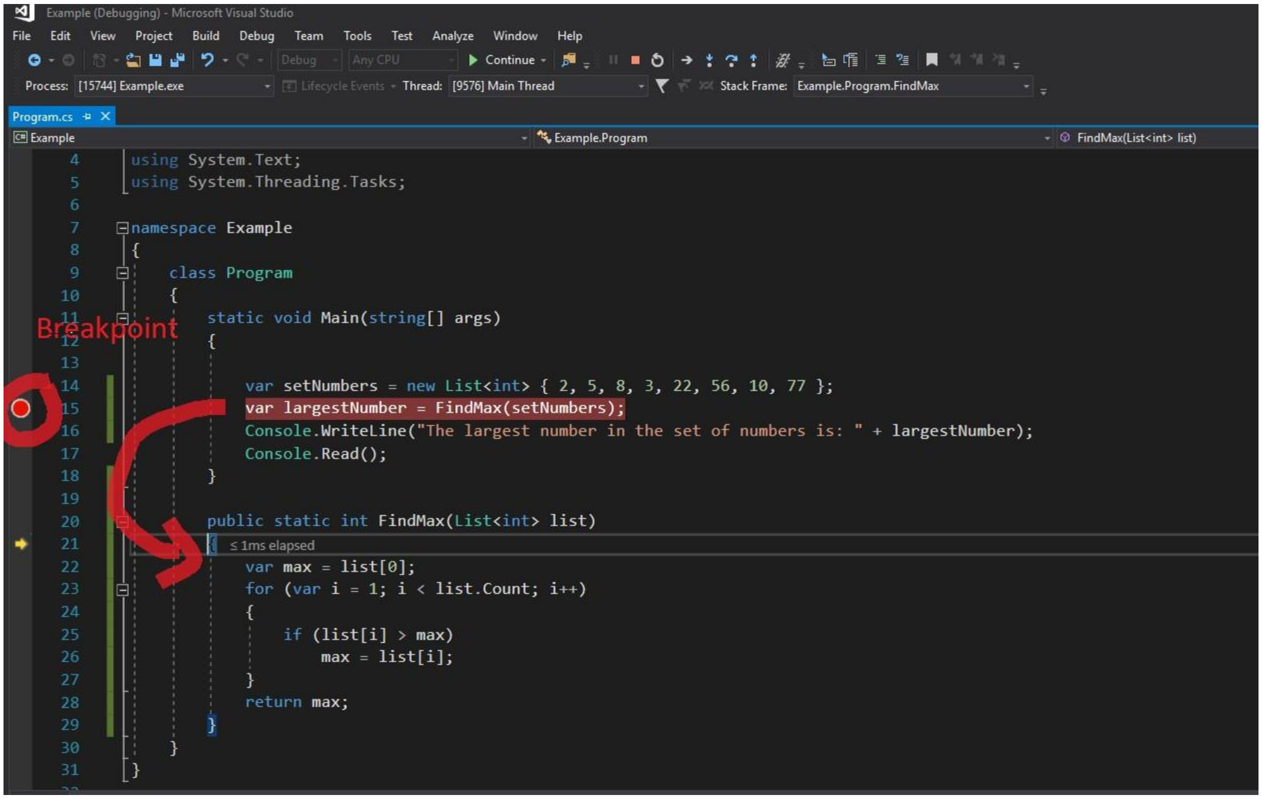
You can hover over the variable to view the current value in a data tip



```
        {

            var setNumbers = new List<int> { 2, 5, 8, 3, 22, 56, 10, 77 };
            var largestNumber = FindMax(setNumbers);
            Console.WriteLine("The largest number in the set of numbers is: " + largestNumber);  ≤ 2ms elapsed
            Console.Read();                                                    largestNumber 77
        }

        public static int FindMax(List<int> list)
        {
            var max = list[0];
            for (var i = 1; i < list.Count; i++)
```

Picture 18 - Result of Step Over in Visual Studio

4. **STEP INTO:** this is like *Step Over* (F10), if the current yellow highlighted section is a method calls, the debugger will *go inside* the method


Example (Debugging) - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
Any CPU
Continue
Lifecycle Events
Program.cs
Breakpoint
≤ 1ms elapsed

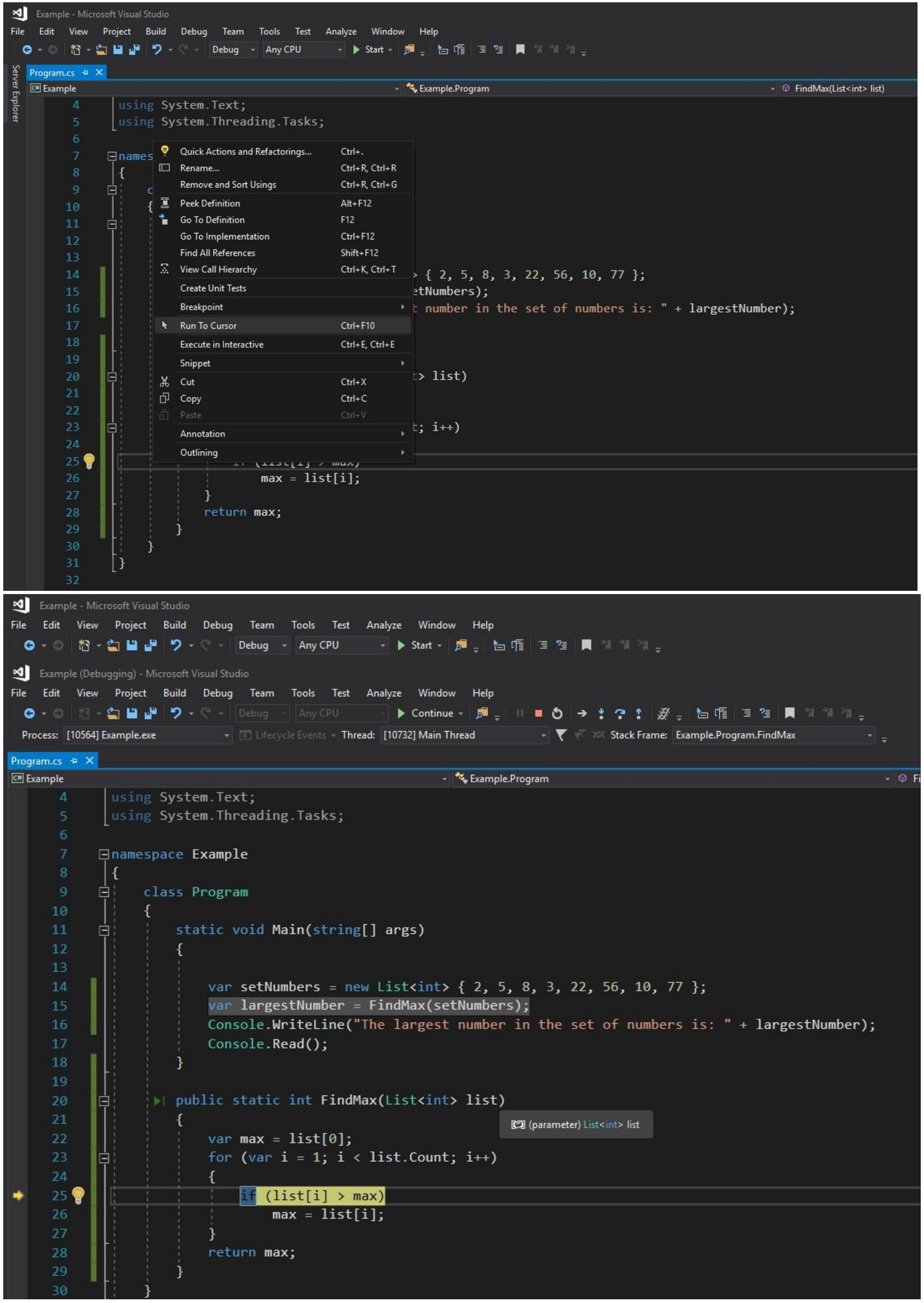
Process: [15744] Example.exe
Thread: [9576] Main Thread
Stack Frame: Example.Program.FindMax
Example
Example.Program
FindMax(List<int> list)

```
using System.Text;
using System.Threading.Tasks;

namespace Example
{
    class Program
    {
        static void Main(string[] args)
        {

            var setNumbers = new List<int> { 2, 5, 8, 3, 22, 56, 10, 77 };
            var largestNumber = FindMax(setNumbers);
            Console.WriteLine("The largest number in the set of numbers is: " + largestNumber);
            Console.Read();
        }

        public static int FindMax(List<int> list)
        {
            var max = list[0];
            for (var i = 1; i < list.Count; i++)
            {
                if (list[i] > max)
                    max = list[i];
            }
            return max;
        }
    }
}
```

Picture 19 - Step into in Visual Studio

5. When you are debugging inside a method and you want to go back the main code. Press Shift – F11 within the current method, and the method will be executed and complete, then it will pause at the line where it called.

6. **RUN TO CURSOR:** This step will start debugging and sets a temporary breakpoint on the current line of code. Chose Stop Debugging or press Shift – F5. In line 25, right-click and chose Run to Cursor or press Ctrl + F10.



Example - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
Any CPU
Start
Program.cs
Example
Example.Program
FindMax(List<int> list)
Server Explorer
Quick Actions and Refactorings... Ctrl+.
Rename... Ctrl+R, Ctrl+R
Remove and Sort Usings Ctrl+R, Ctrl+G
Peek Definition Alt+F12
Go To Definition F12
Go To Implementation Ctrl+F12
Find All References Shift+F12
View Call Hierarchy Ctrl+K, Ctrl+T
Create Unit Tests
Breakpoint
Run To Cursor Ctrl+F10
Execute in Interactive Ctrl+E, Ctrl+E
Snippet
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Annotation
Outlining

```
 4   using System.Text;
 5   using System.Threading.Tasks;
 6
 7   names
 8   {
 9       c
10       {
11
12
13
14   > { 2, 5, 8, 3, 22, 56, 10, 77 };
15   etNumbers);
16   t number in the set of numbers is: " + largestNumber);
17
18
19
20   t> list)
21
22
23   t; i++)
24
25
26                   max = list[i];
27               }
28               return max;
29           }
30       }
31   }
32
```

Example - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
Any CPU
Start
Example (Debugging) - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
Any CPU
Continue
Process: [10564] Example.exe
Lifecycle Events
Thread: [10732] Main Thread
Stack Frame: Example.Program.FindMax
Program.cs
Example
Example.Program
(parameter) List<int> list

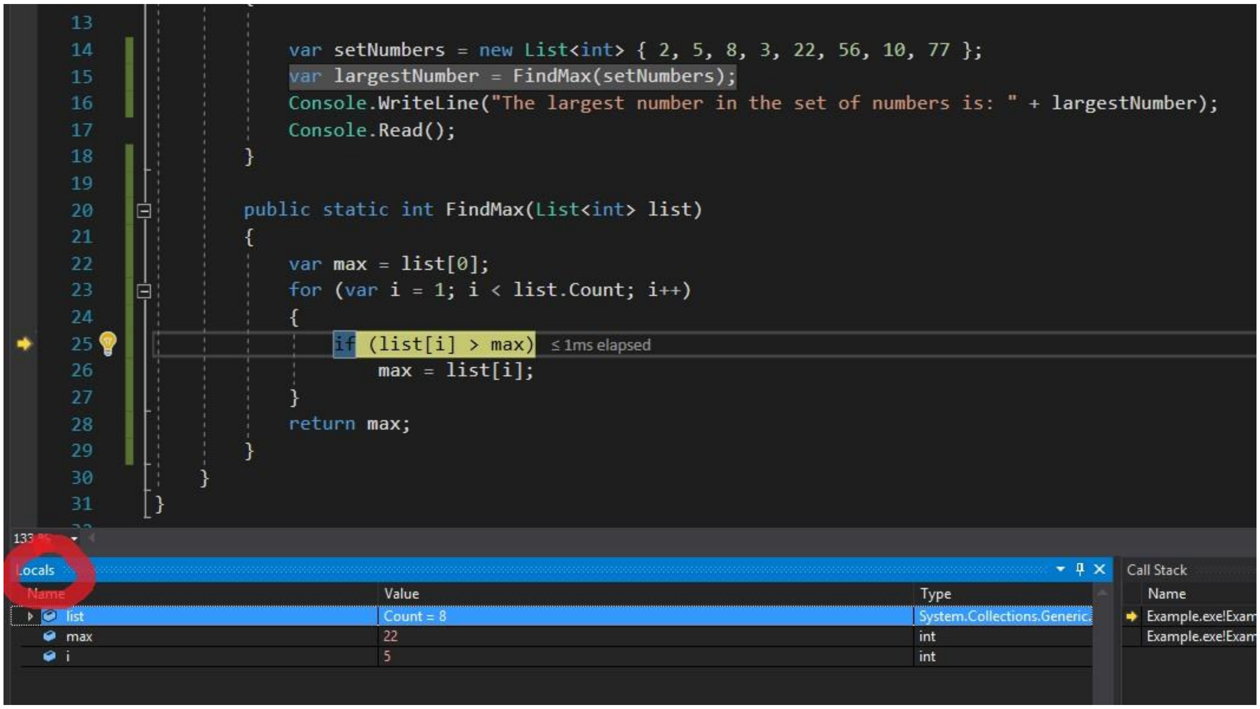
```
 4   using System.Text;
 5   using System.Threading.Tasks;
 6
 7   namespace Example
 8   {
 9       class Program
10       {
11           static void Main(string[] args)
12           {
13
14               var setNumbers = new List<int> { 2, 5, 8, 3, 22, 56, 10, 77 };
15               var largestNumber = FindMax(setNumbers);
16               Console.WriteLine("The largest number in the set of numbers is: " + largestNumber);
17               Console.Read();
18           }
19
20           public static int FindMax(List<int> list)
21           {
22               var max = list[0];
23               for (var i = 1; i < list.Count; i++)
24               {
25                   if (list[i] > max)
26                       max = list[i];
27               }
28               return max;
29           }
30       }
```

Picture 20 - Run to cursor in Visual Studio

This command is useful when you are editing code and want to set a temporary breakpoint quickly and start the debugger.

7. INSPECT VARIABLES:

- The **Locals**: this window displays variables that appear in the local scope

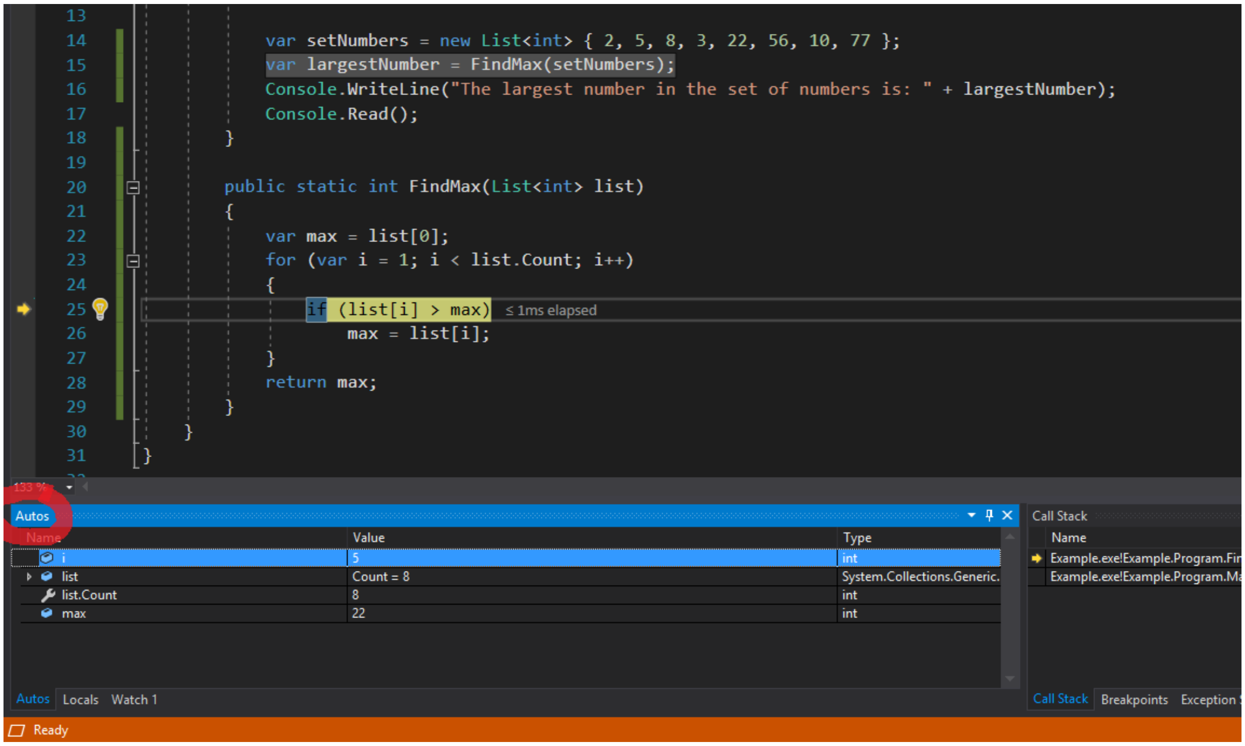


```
            var setNumbers = new List<int> { 2, 5, 8, 3, 22, 56, 10, 77 };
            var largestNumber = FindMax(setNumbers);
            Console.WriteLine("The largest number in the set of numbers is: " + largestNumber);
            Console.Read();
        }

        public static int FindMax(List<int> list)
        {
            var max = list[0];
            for (var i = 1; i < list.Count; i++)
            {
                if (list[i] > max)  ≤ 1ms elapsed
                    max = list[i];
            }
            return max;
        }
    }
}
```

Name	Value	Type
list	Count = 8	System.Collections.Generic.
max	22	int
i	5	int

Picture 21 - Inspect variables

- The **Autos**: this window displays variables are being used around the current line.



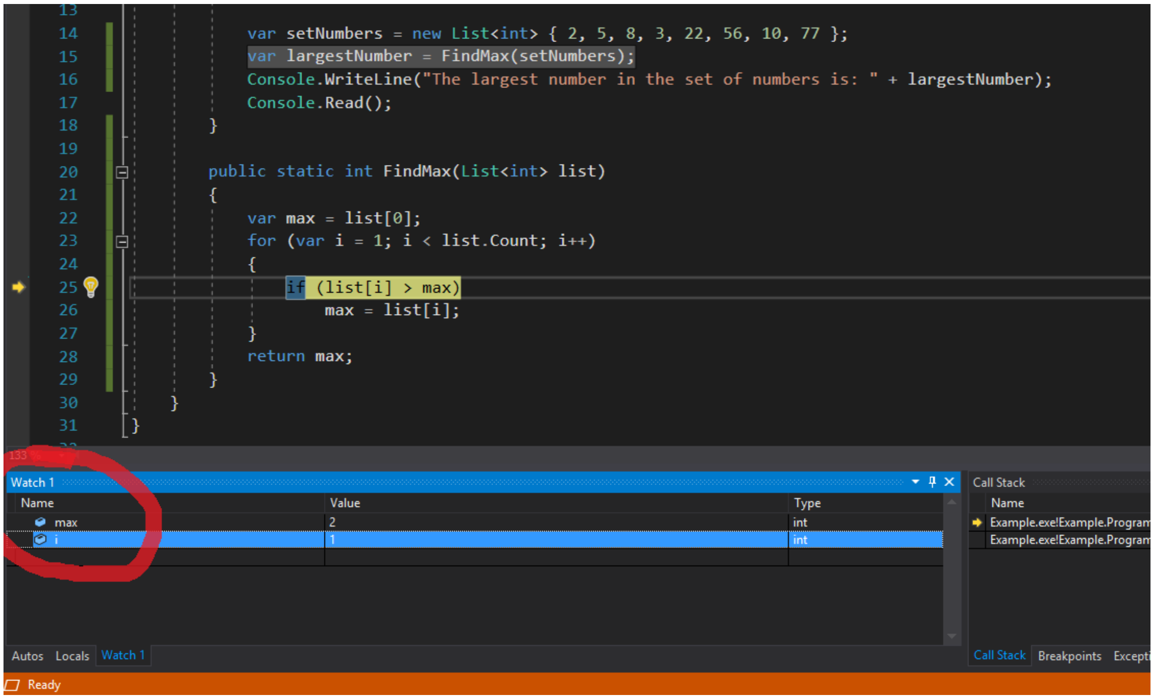
```
            var setNumbers = new List<int> { 2, 5, 8, 3, 22, 56, 10, 77 };
            var largestNumber = FindMax(setNumbers);
            Console.WriteLine("The largest number in the set of numbers is: " + largestNumber);
            Console.Read();
        }

        public static int FindMax(List<int> list)
        {
            var max = list[0];
            for (var i = 1; i < list.Count; i++)
            {
                if (list[i] > max)
                    max = list[i];
            }
            return max;
        }
    }
}
```

Name	Value	Type
i	5	int
list	Count = 8	System.Collections.Generic.
list.Count	8	int
max	22	int

Name
Example.exe!Example.Program.Fi
Example.exe!Example.Program.M

Picture 22- The Autos

- You can also use the **Watch** (Debug/ Windows/ Watch/ Watch (1,2,3,4)) and **QuickWatch** (right-click on variable / Debug/ QuickWatch) to inspect the variables.



```
            var setNumbers = new List<int> { 2, 5, 8, 3, 22, 56, 10, 77 };
            var largestNumber = FindMax(setNumbers);
            Console.WriteLine("The largest number in the set of numbers is: " + largestNumber);
            Console.Read();
        }

        public static int FindMax(List<int> list)
        {
            var max = list[0];
            for (var i = 1; i < list.Count; i++)
            {
                if (list[i] > max)
                    max = list[i];
            }
            return max;
        }
    }
}
```

Watch 1

Name	Value	Type
max	2	int
i	1	int

Call Stack

Name
Example.exe!Example.Program
Example.exe!Example.Program

Picture 23 - The Watch

5. Conclusion:

Debugging is the art of diagnosing errors in program and then you will know how to correct them. I find it is very important to learn how to debug a program so you can make an application that is more secure and more effective.

No discussion of debugging helps you to prevent bugs. It doesn’t matter how well you write code, if you write a wrong program, it won’t help anyone. If you write a correct program, but the user cannot use it, so it would count as you haven’t written the code. So a good debugger will let them know where the problem is.

In short, I will show some basic debugging steps:

- Recognize if the bug appears.
- Isolate source of bug.
- Identify cause of bug.
- Determine how to fix the bug.
- Fix and test.

In this section, I have shown you how to use debug tools in Visual Studio 2017 and their debugging facilities in details, they are very powerful feature and there are close step of debugging. In conclusion, debugging will help programmer to develop a program more secure, robust applications

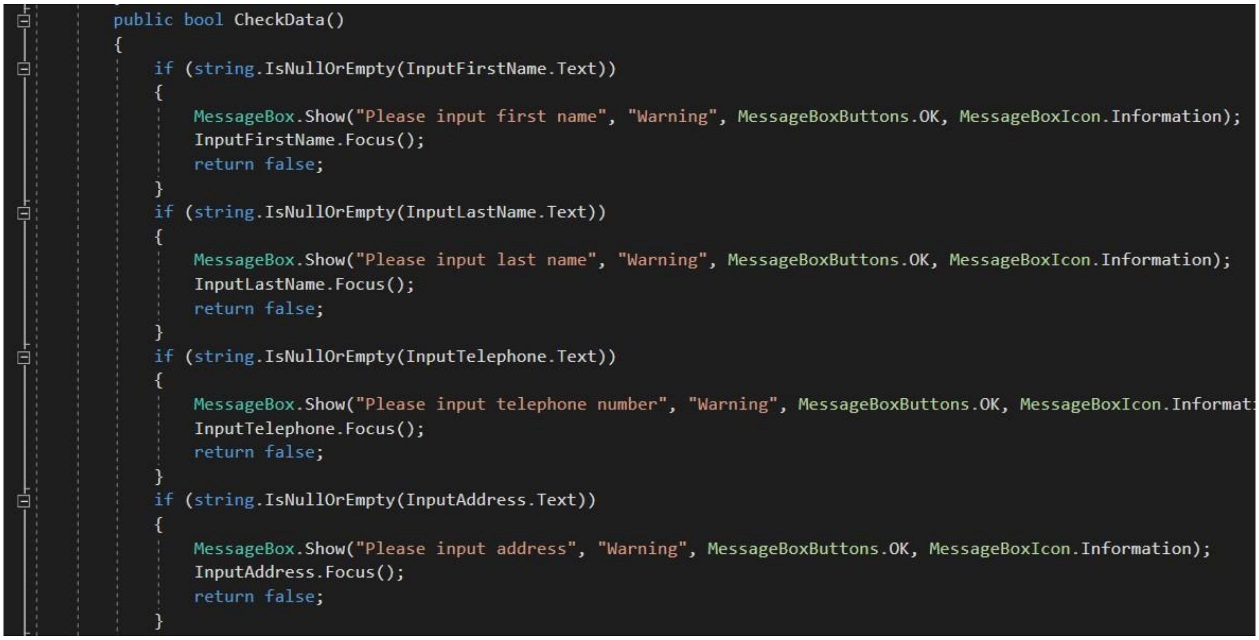
Part4: Coding Standard

I. Definition

Coding standards define a programming style. This is a set of rules and guidelines for the formatting of source code. Such as naming class, properties or methods; declaring variable; comments, documentation etc. (sitepoint, n.d.)

II. Some common aspects of coding standard:

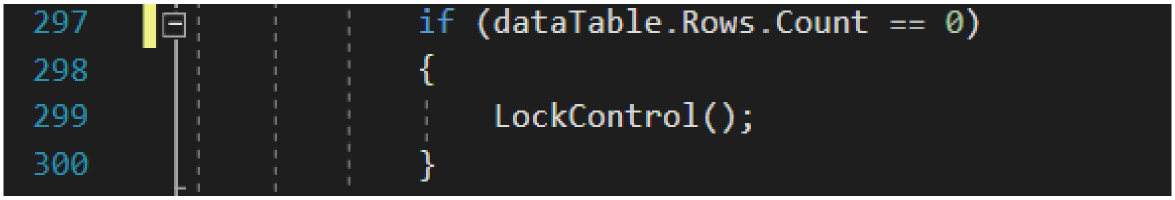
- Naming: Rules of capitalization
  - Pascal case: capitalize the first character of all word and other character are lower case. Use for class and method names.
  - Camel case: same as Pascal case but lowercase the first letter. Use for variables and method parameters.
- Curly brackets: these symbols must be put under the line if, for, while etc.



```
public bool CheckData()
{
    if (string.IsNullOrEmpty(InputFirstName.Text))
    {
        MessageBox.Show("Please input first name", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Information);
        InputFirstName.Focus();
        return false;
    }
    if (string.IsNullOrEmpty(InputLastName.Text))
    {
        MessageBox.Show("Please input last name", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Information);
        InputLastName.Focus();
        return false;
    }
    if (string.IsNullOrEmpty(InputTelephone.Text))
    {
        MessageBox.Show("Please input telephone number", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Informat
        InputTelephone.Focus();
        return false;
    }
    if (string.IsNullOrEmpty(InputAddress.Text))
    {
        MessageBox.Show("Please input address", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Information);
        InputAddress.Focus();
        return false;
    }
```

Picture 24 - Example of coding standard

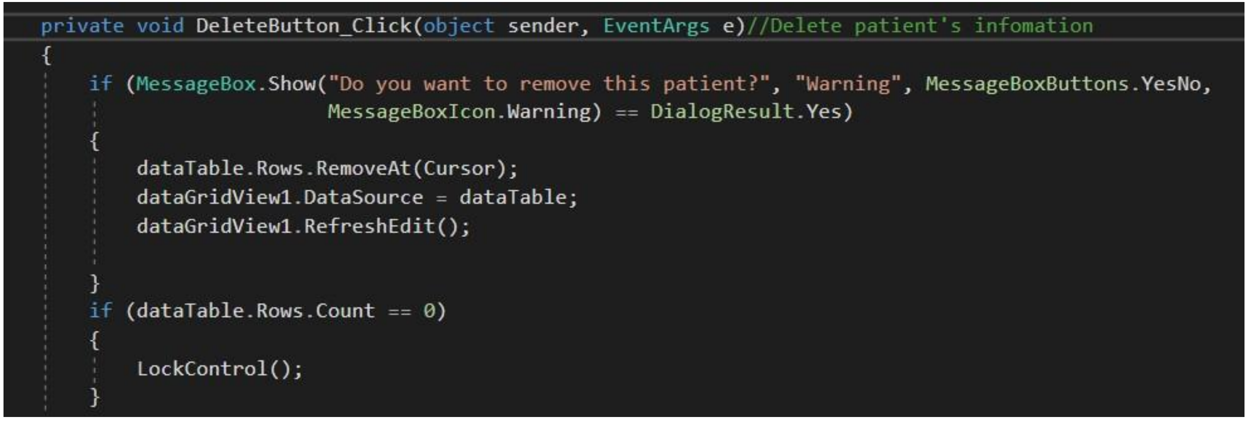
- Inter-term spacing: Put space after commas and semi-colon, around operators



```
if (dataTable.Rows.Count == 0)
{
    LockControl();
}
```

Picture 25 - Put space after commas

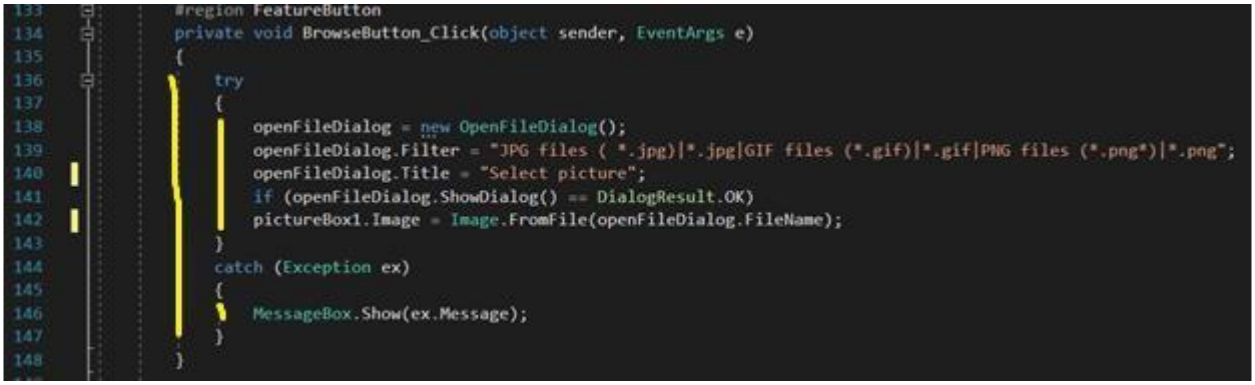
- Comment: Comment need to be simple as possible and never write comment that repeat the code.



```
private void DeleteButton_Click(object sender, EventArgs e)//Delete patient's infomation
{
    if (MessageBox.Show("Do you want to remove this patient?", "Warning", MessageBoxButtons.YesNo,
                        MessageBoxIcon.Warning) == DialogResult.Yes)
    {
        dataTable.Rows.RemoveAt(Cursor);
        dataGridView1.DataSource = dataTable;
        dataGridView1.RefreshEdit();

    }
    if (dataTable.Rows.Count == 0)
    {
        LockControl();
    }
```

Picture 26 - Comment in IDE

- Make lines at same level of nesting stand out

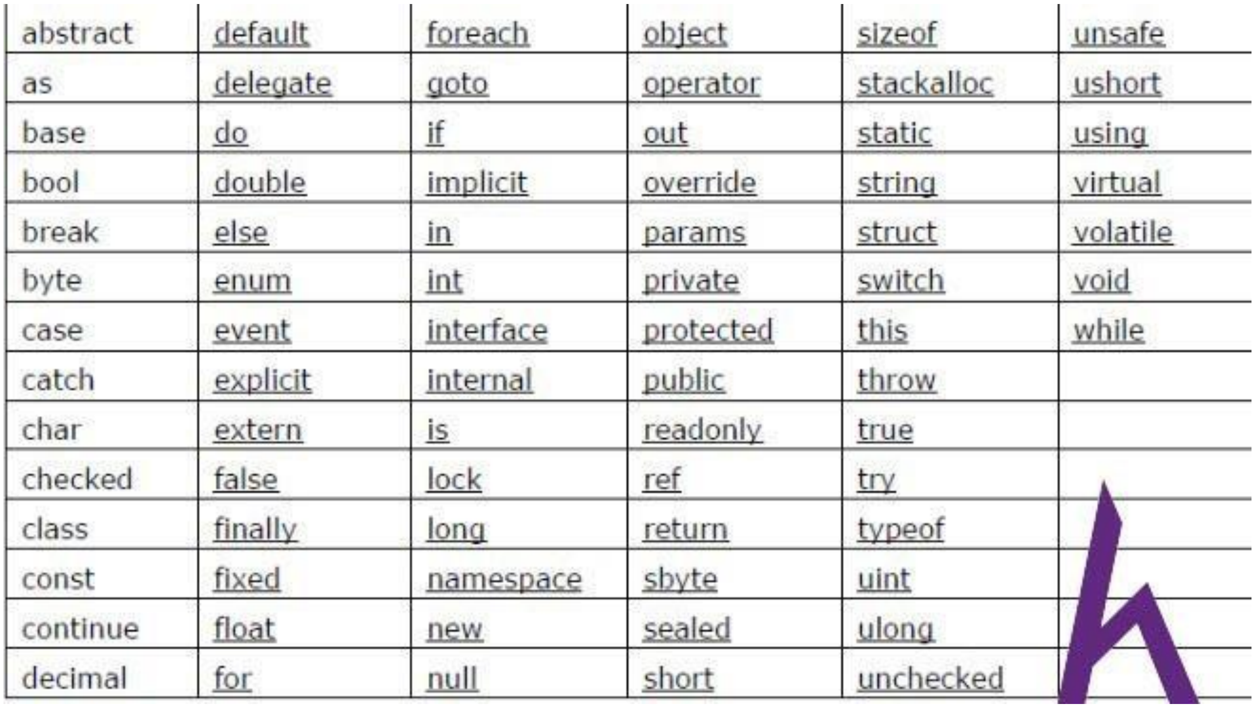


```
#region FeatureButton
private void BrowseButton_Click(object sender, EventArgs e)
{
    try
    {
        openFileDialog = new OpenFileDialog();
        openFileDialog.Filter = "JPG files ( *.jpg)|*.jpg|GIF files (*.gif)|*.gif|PNG files (*.png*)|*.png";
        openFileDialog.Title = "Select picture";
        if (openFileDialog.ShowDialog() == DialogResult.OK)
        pictureBox1.Image = Image.FromFile(openFileDialog.FileName);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Picture 27 - Make lines at the same level

Blank line: The lines of code that are related to each other should be put in one block. Two blocks of code should be separated by a blank line

- Methods:
  - Don’t name unclearly.
  - Never distinguish the methods by number.
- Rules for name variables:
  - Variable name is a string of linked characters (no spaces) and no special characters.
  - Variable names must not be placed in accented Vietnamese.
  - Names cannot start with numbers.
  - Variable names cannot be identical.
  - Variable names cannot match keywords:
    - □ List of keywords in c#:



abstract	default	foreach	object	sizeof	unsafe
as	delegate	goto	operator	stackalloc	ushort
base	do	if	out	static	using
bool	double	implicit	override	string	virtual
break	else	in	params	struct	volatile
byte	enum	int	private	switch	void
case	event	interface	protected	this	while
catch	explicit	internal	public	throw	
char	extern	is	readonly	true	
checked	false	lock	ref	try	
class	finally	long	return	typeof	
const	fixed	namespace	sbyte	uint	
continue	float	new	sealed	ulong	
decimal	for	null	short	unchecked	

Table 1 - List of keywords in C#. (howkteam, n.d.)

III. Coding standard is necessary in a team as well as for the individual:

Think of your main product and how it evolved. Each month, you ship new features. Each quarter, you comb through bug reports. And then look for ideas to improve. Each developer on the team has their style. For instance, one developer may religiously use comments to describe what they're doing. Another developer may use the "write drunk, edit sober" philosophy and focus on achieving quality through debugging. As a manager or team lead, it's tough to stay organized and productive. (plesk, n.d.)

Widely differing code standards also hurt your team's effectiveness and morale in subtle ways. Think about hiring new people to the team. How long will it take new hires to learn how your team codes? Without a clear commitment to a coding standard, the onboarding process will take longer. Furthermore, when people use different coding standards, debugging and testing become more difficult.

Ask 10 developers, and you'll get 11 answers. Rather than prescribe a solution for your company, we suggest reviewing the following options first. But a word of caution: It's easy to go overboard with coding standards. A small number of rules or principles will go a long way.

- Issues to consider when creating coding standards:
  - Emphasize decision making values: Principles help guide how a developer approaches code.
    - Example: The agile emphasis on close collaboration with the business helps you plan your daily coding work.
  - Prioritize the long term: A principles-based coding standard should be able to stand for two or three years without any changes.
- The procedures approach: Coding standards with all the details:
  - Choose one focus language: Your company may use JavaScript, C# and PHP. Covering all of these in your coding standards may not be practical.
  - Balance theory and practice: For procedures to be beneficial, state a brad principle and provide examples of good code.
    - Example: Write comments in clear English with minimal slang.
  - Set a length limit: A 200-page coding standard document, even if valuable, won't encourage developers to open the document.
  - Variable names cannot be identical.
- Coding standards tips:
  - If you use the procedure approach to coding standards, appoint someone to own and maintain the standards. Otherwise, you may end up with an out of date document that provides little value.
  - Now based on those options, make some choices about what makes sense for your company. Balance your personal preferences with what your customers value. For instance, national defense customers will place a high value on security and internal control. If that's what your customers want, then start building those points into your product.

References

c4learn. (n.d.). Retrieved from c4learn: http://www.c4learn.com/c-programming/c-local-variable-scope/

ee. (n.d.). Retrieved from ee: http://www.ee.ncu.edu.tw/~jfli/soc/lecture/Lab_02.pdf

howkteam. (n.d.). Retrieved from howkteam: https://www.howkteam.vn/course/cau-truc-lenh-co-ban-trong-c-console-application/bien-trong-c-52

lynda. (n.d.). Retrieved from lynda: https://www.lynda.com/C-tutorials/Learning-C-Algorithms/604241- 2.html

myassignmenthelp. (n.d.). Retrieved from myassignmenthelp: https://www.myassignmenthelp.net/uk/btec/hnd-enhancing- professional-practices

plesk. (n.d.). Retrieved from plesk: https://www.plesk.com/blog/tips-easy-reading/coding-standards- matter/

sitepoint. (n.d.). Retrieved from sitepoint: https://www.sitepoint.com/coding-standards/

wikipedia. (n.d.). Retrieved from wikipedia: https://en.wikipedia.org/wiki/Integrated_development_environment