Lab 01:

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG C#

Với ứng dụng CONSOLE

A. MỤC TIÊU:

- ✓ Hướng dẫn sinh viên làm quen với ngôn ngữ lập trình C#: qua việc viết các ứng dụng **Console** trong vs. NET 2015/2017/ 2019.
- ✓ Xây dựng các lớp, tạo đối tượng, truy xuất các phương thức, …
- ✓ Soạn thảo mã nguồn, biên dịch, debug, thực thi chương trình…
- ✓ Kế thừa trong lập trình hướng đối tượng trên C#.
- ✓ Tìm hiểu về sử dụng ***lambda expression*** trong thư viện **LINQ** của .NET
- ✓ Khuyến khích sinh viên sử dụng chuẩn viết code C# (***C# Coding Convetion***)
- ✓ Ngôn ngữ C#

- Cấu trúc chương trình C#: Namespace, class (thuộc tính, phương thức), Comment
- Kiểu dữ liệu: bool, decimal, double, float, int, string, `DateTime`

bool?, decimal?, double?, float?, int?, long?, `DateTime?`...

Nullable<bool>, Nullable<decimal>, Nullable<double>, Nullable<float>, Nullable<int>…

object, var, dynamic

- Chuyển đổi kiểu dữ liệu: Ép kiểu (int), as, **Parse**, Convert
- Toán tử trong C#: +, -, *, /, %, ++, --…
- Lệnh If, Else và toán tử Exp1 ? Exp2 : Exp3;
- Vòng lặp: `for`, `foreach`, `while`, `do`.. `while` và Lệnh điều khiển `break`, `continue`, `return`
- Tính đóng gói: bởi sử dụng Access Specifier:,`private`, `protected`, `internal`, `protected internal`, `public`
- Tính kế thừa: để tái sử dụng code giúp thời gian thực thi nhanh hơn

  class <derived_class> : <base_class>
- Coding Convention C#: Đưa ra các quy ước khi coding với ngôn ngữ lập trình C#, với các quy tắc này giúp tiết kiệm thời gian rất trong quá trình phát triển và bảo trì phần mềm.

  Ví dụ: Đặt tên class dùng danh từ, đối với phương thức dùng động từ …

  Tên biến, tên phương thức thể hiện được ý nghĩa

  Nên comment những đoạn code khó hiểu hoặc có chức năng đặc biệt
- Một số phương thức của thư viện Console

```
Console.Write(<giá trị cần in ra màn hình>);
Console.WriteLine();   // Sử dụng lệnh in ra màn hình có xuống dòng
Console.Read();        //đọc 1 ký tự và trả về 1 số nguyên là mã ASCII của ký tự đó
```

```
Console.ReadLine();     / / Đọc dữ liệu chuỗi từ bàn phím cho đến khi gặp ký tự xuống dòng
Console.ReadKey();      // Dừng màn hình để xem kết quả.
```

B. BÀI TẬP

Bài tập 1A: Tạo lớp Student có các dữ liệu và phương thức sau

- Mã số sinh viên
- Họ Tên sinh viên
- Điểm trung bình học tập
- Khoa

  + Tạo Property cho các thành viên trên

  + Tạo constructor không có tham số, và có tham số

Tạo lớp **Tester**, trong lớp này chỉ chứa duy nhất hàm **main()**. Hàm này cho phép người dùng nhập vào số **N** là tổng số sinh viên, sau đó lần lượt tạo các đối tượng sinh viên và đưa vào mảng các **Student** theo những thông tin do user nhập vào (dùng vòng lặp for). Cuối cùng xuất ra danh sách chi tiết thông tin sinh viên.

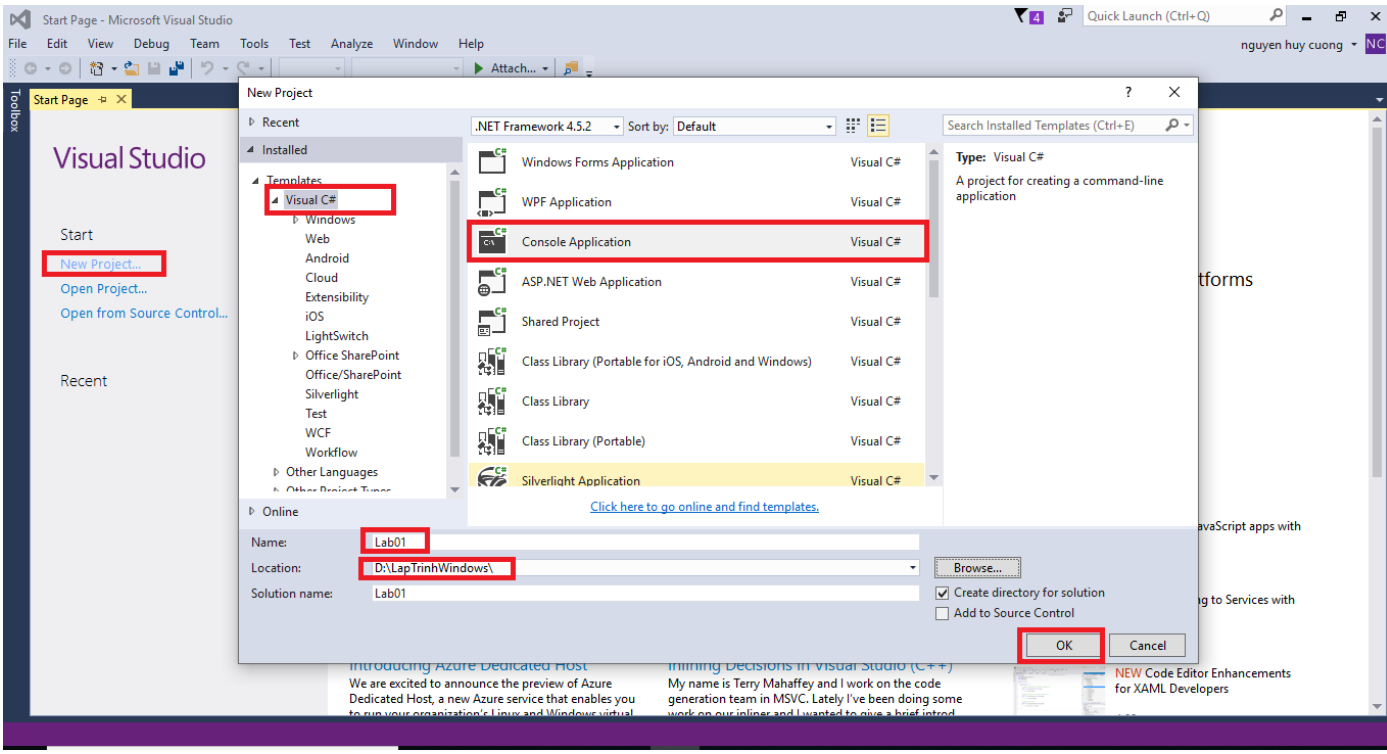
Yêu cầu:

- ✓ Sinh viên xây dựng chương trình theo nội dung mô tả bên trên.
- ✓ Compile & Build chương trình.
- ✓ Run chương trình ở hai chế độ debug và không debug.
- ✓ Chạy từng bước chương trình trong chế độ debug: dùng breakpoint hoặc chạy từng dòng lệnh. Kiểm tra những giá trị của các biến trong chương trình ở cửa sổ Watch.

Hướng dẫn:

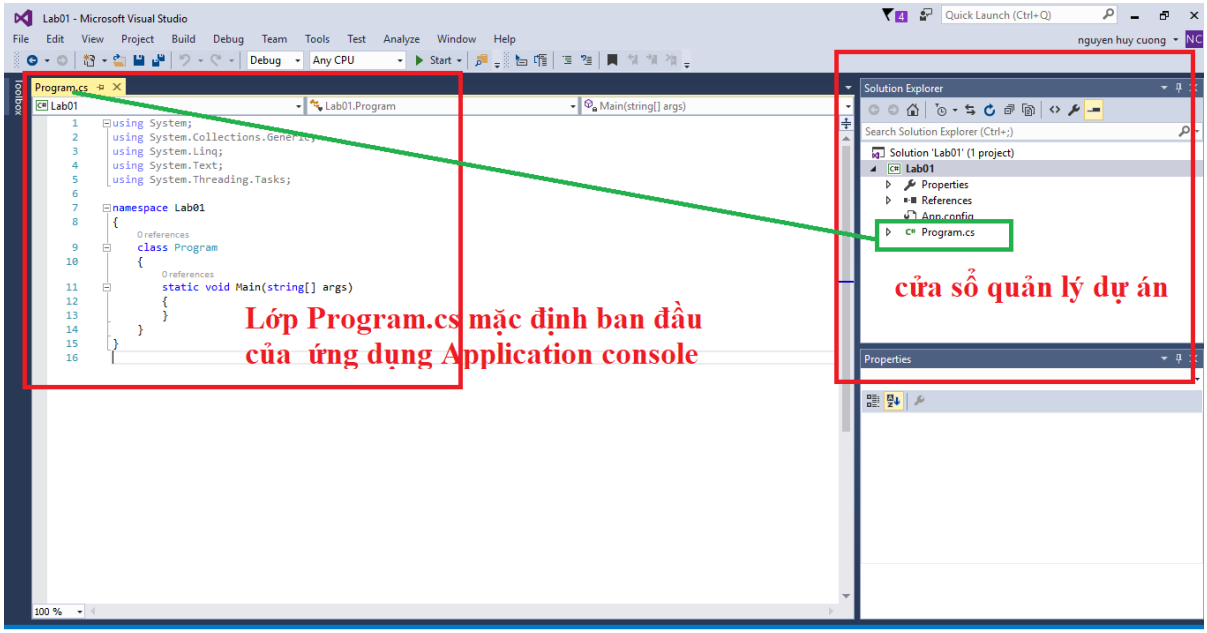
Bước 1*: Tạo project mới trong VS 2015*

- Mở Visual Studio 2015, chọn **New Project**, Chọn ngôn ngữ **Visual C#**, Loại project là **Console Application**
- Đặt tên project là ***Lab01*** và lưu ở Folder ***D:\LapTrinhWindows***



Hình 1: Tạo một project console c# mới trong VS .NET 2015

- Click OK để đồng ý tạo project, kết quả chúng ta được một ứng dụng console như sau:



Hình 2: Màn hình làm việc của Project

- Lưu lại dự Án File/Save All

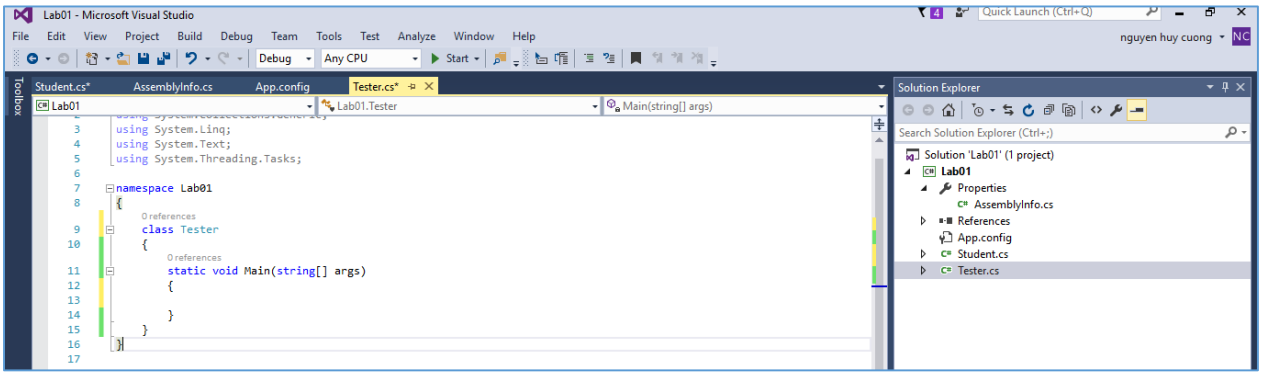
Bước 2: *Viết yêu cầu tạo class Student, Khai báo thuộc tính, phương thức, SV tự làm hoàn chỉnh theo yêu cầu của bài tập 1*

```
class Student
    {
        //1. Tao thuoc tinh
        private string studentID;
```

```
        // Lam tuong tu cho cac thuoc tinh con lai
        //
        //
        //
        //2. Tao cac Property
        public string StudentID
        {
            get
            {
                return studentID;
            }
            set
            {
                studentID = value;
            }
        }
        // 3. Tao constructor mac dinh, tham so
        public Student()
        {

        }
        // 4. Tao constuctor co tham so
        public Student(string id)
        {
            ////
        }
    }
```

***Bước 3**: Xóa lớp Program, tạo lớp Tester theo yêu cầu của bài tập*

➔ *Hoặc đổi tên Rename lớp Program cũng tương tự*



Hình 3: Tạo chương trình với 2 lớp Tester.cs và Student.cs

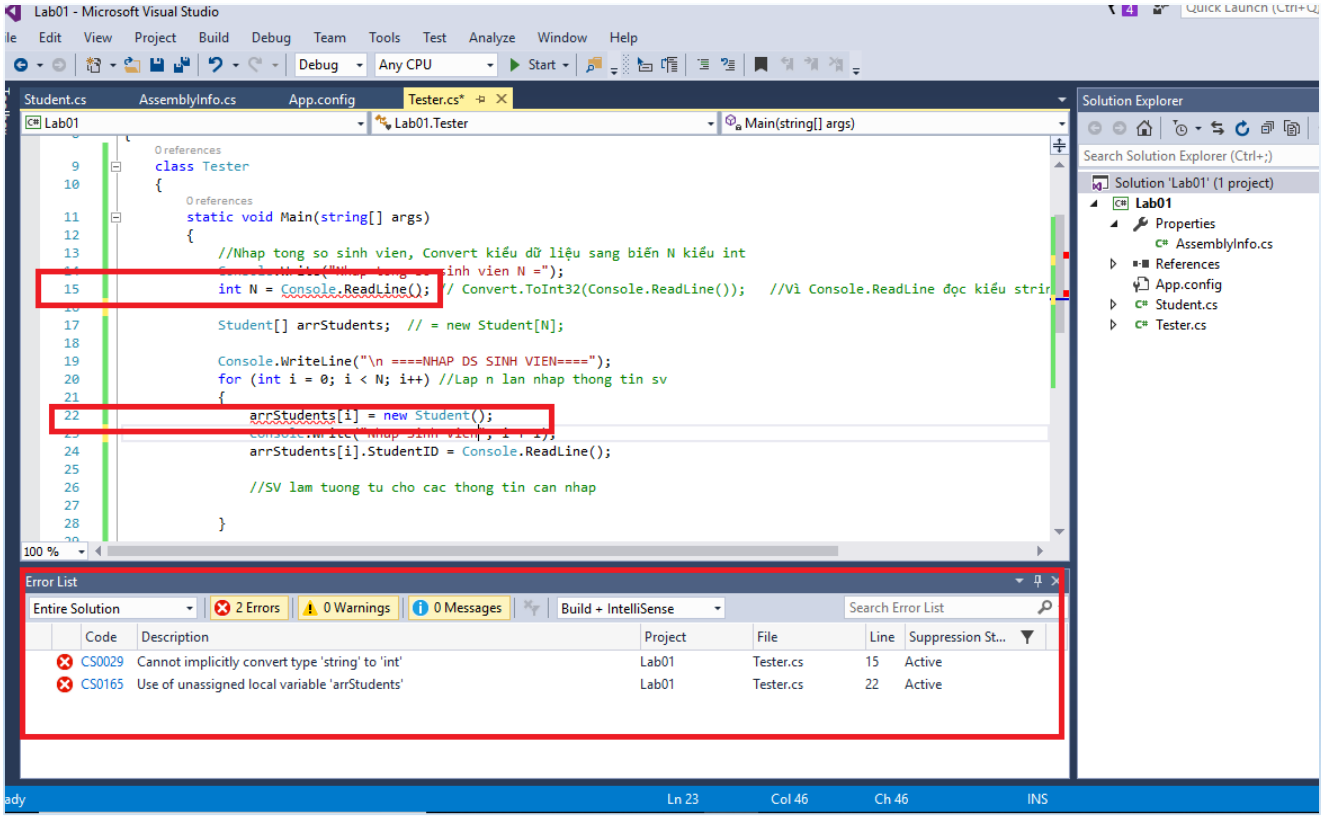
- Viết code trong hàm **Main()** để cho phép người dùng nhập vào tổng số sinh viên

```
static void Main(string[] args)
{
                        //Nhap tong so sinh vien, Convert kiểu dữ liệu sang biến N
            kiểu int
                        Console.Write("Nhap tong so sinh vien N =");

                        int N = Convert.ToInt32(Console.ReadLine());
                        Student[] arrStudents = new Student[N];

                        Console.WriteLine("\n ====NHAP DS SINH VIEN====");
                        for (int i = 0; i < N; i++) //Lap n lan nhap thong tin sv
                        {
                            arrStudents[i] = new Student();
                            Console.Write("Nhap MaSV {0}:", i + 1);
```

```
                    arrStudents[i].StudentID = Console.ReadLine();

                    //SV lam tuong tu cho cac thong tin can nhap

                }

                //Xuat DS Sinh vien
                Console.WriteLine("\n ====XUAT DS SINH VIEN====");
                foreach (Student sv in arrStudents)
            {
                    Console.WriteLine("MSSV:{0}", sv.StudentID);
            }

                //De dung man hinh cho dễ kiểm tra kq
                 Console.ReadLine();
    }
```

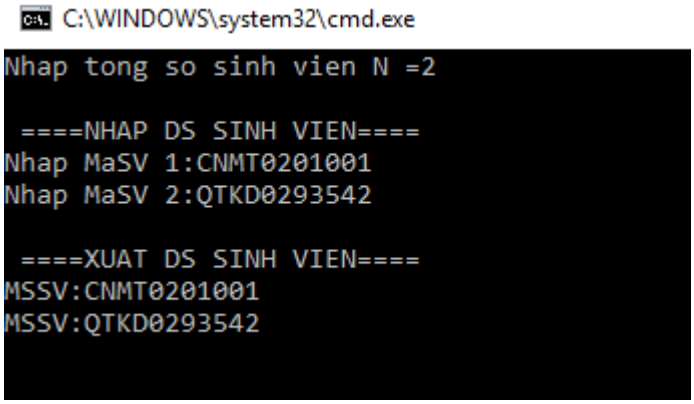
Bước 4: *Biên dịch và chạy chương trình*

- ✓ Để biên dịch chương trình chọn menu ***Build,*** rồi chọn ***Build Solution (***hoặc dùng phím tắt **F6** hoặc click phải vào Solution Explorer chọn **Build).** VS.NET sẽ thông báo biên dịch thành công hay gặp lỗi cú pháp.



Hình 4: VS thông báo tổng số lỗi, số dòng bị lỗi khi viết code

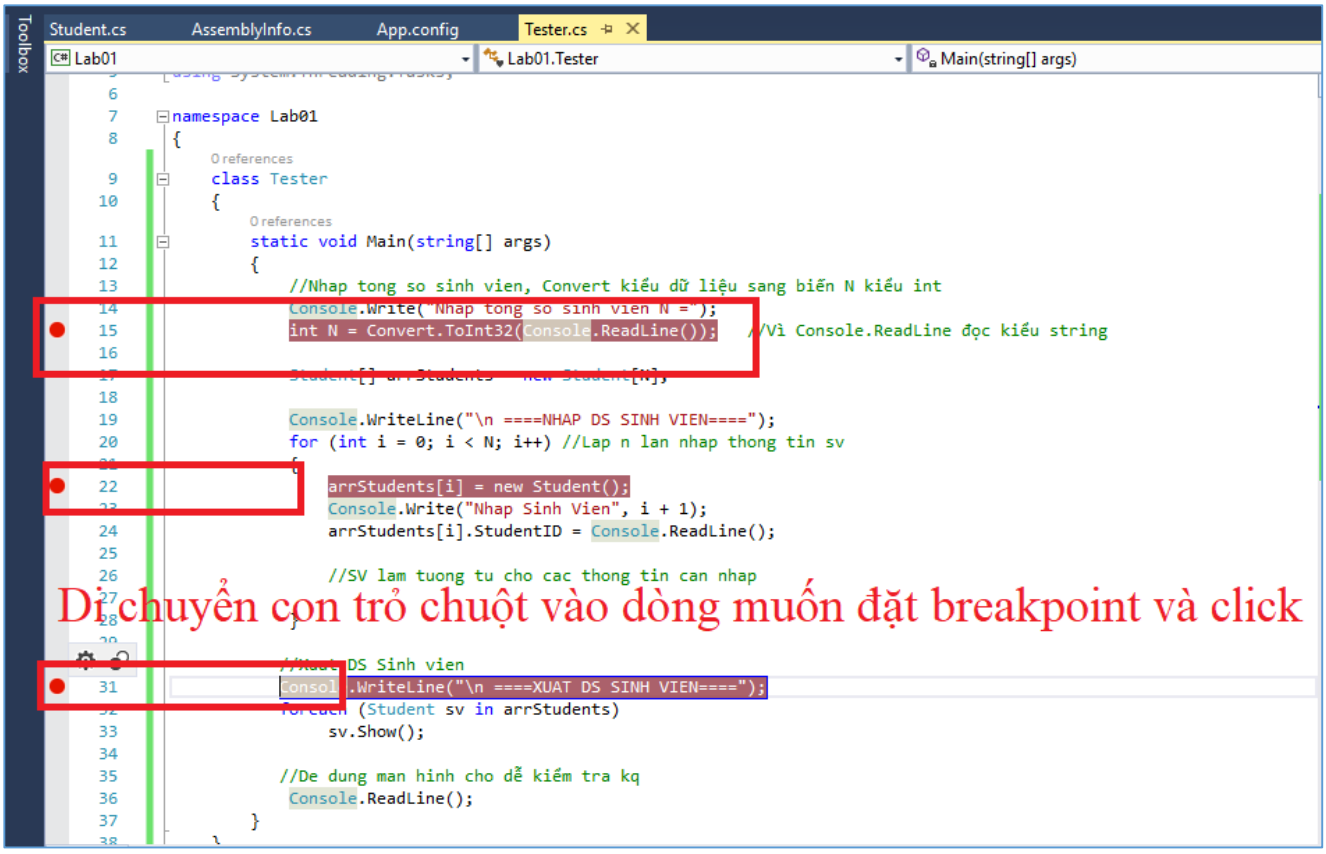
- ✓ Để chạy chương trình click vào biểu tượng **Start** trên VS (hoặc chọn **Debug** / ***Debug Without Debugging*** hoặc dùng ***Ctrl +F5)***



Hình 5: Kết quả màn hình console chương trình khi thực hiện

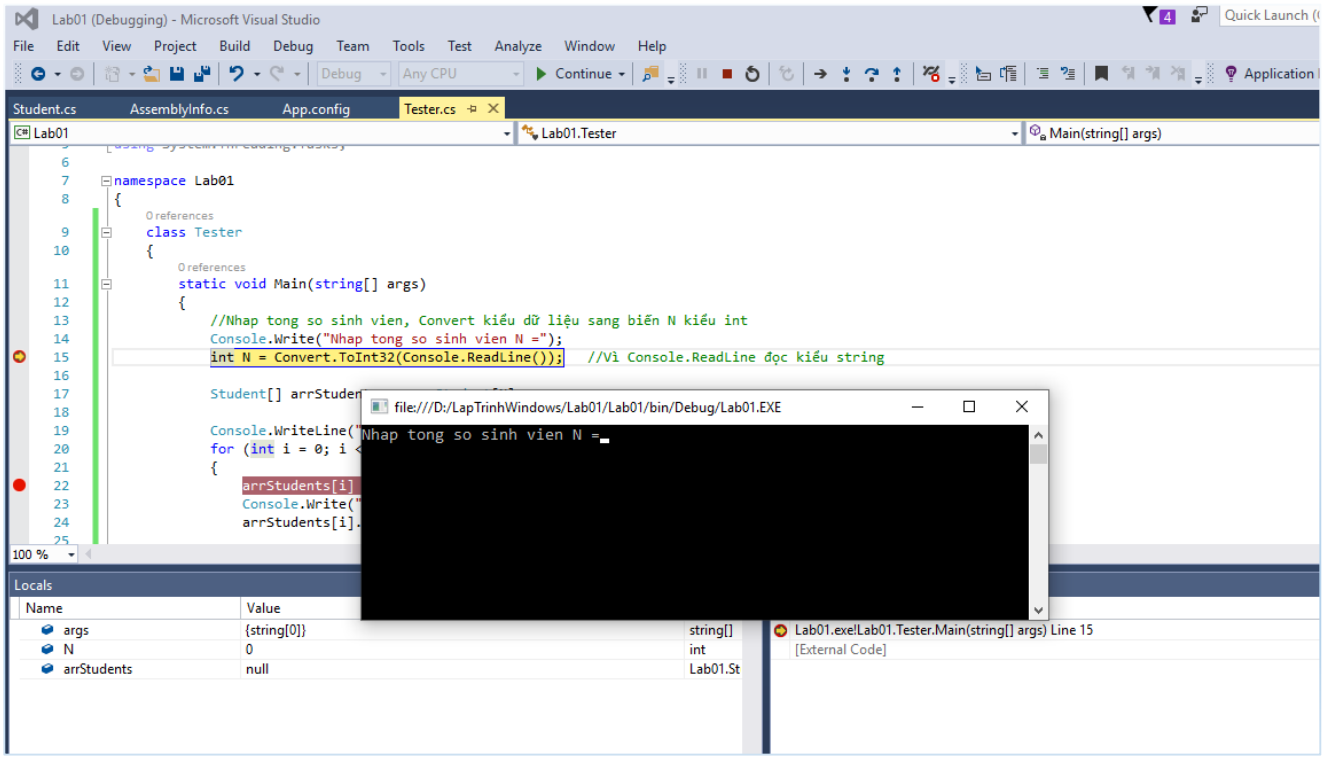
✓ Để bebug chương trình sử dụng F5(hoặc vào Debug / Start Debugging) kết hợp với việc đặt các breakpoint

- Tạo breakpoint bằng cách đơn giản nhất là click chuột vào đầu dòng code (như trong hình). Để huỷ breakpoint, chỉ cần click chuột vào breakpoint đó một lần nữa. Ngoài ra các bạn cũng có thể tạo/huỷ breakpoint bằng phím F9.



Hình 6: Đặt các breakpoint hỗ trợ quá trình Debug

- Nhấn F5 để bắt đầu Debug, Tại các vị trí BreakPoint chương trình sẽ dừng lại. Muốn kiểm tra được sự thay đổi giá trị của các biến.



- Sử dụng Step Over / Step Into / Step Out trong quá trình Debug chương trình

Step Over (F10): Chạy step by step, lướt qua hàm (chỉ nhận giá trị return của hàm).

Step Into (F11): Chạy step by step, đi vào nội dung của các hàm con.

Step Out (continue): "Nhảy" đến breakpoint kế tiếp. Nếu không còn breakpoint nào thì sẽ kết thúc debug. Ngoài ra nó còn có chức năng chạy lướt qua hàm con hiện tại.

Bài tập 1B:

- ✓ Không viết lệnh nhập và xuất danh sách sinh viên trực tiếp trong hàm **Main**() mà hãy viết hai phương thức nhập và xuất thông tin sinh viên ở lớp **Student**. Sau đó trong hàm Main() gọi các hàm nhập xuất.
- ✓ Sử dụng collection là **List** để chứa danh sách sinh viên thay thế cho mảng sinh viên trong bài tập 1.
- ✓ Chạy lại kết quả chương trình theo yêu cầu trên.
- ✓ Viết tiếp chương trình thực hiện một số yêu cầu sau. (*Khuyến khích sinh viên viết ra các hàm cho dễ quản lý, những phần chung nên gom vào một hàm*)

- Xuất ra thông tin của các SV đều thuộc khoa "CNTT" (nếu có)

- Xuất ra danh sách các Sinh viên có điểm trung bình cao nhất và thuộc khoa "CNTT". Nếu không có thông báo không tìm được SV.

- Xuất ra danh sách các sinh viên theo thứ tự có điểm trung bình tăng dần. Yêu cầu sử dụng

hàm **sort**() của **List<Student>.**

Hướng dẫn: Để có thể sắp xếp được khi sử dụng hàm **Sort**() ta cần phải Implement phương thức so sánh khi được kế thừa từ Interface **IComparable**

- Cho lớp **Student** kế thừa từ interface `IComparable`

```
class Student : IComparable
```

- Implement phương thức so sánh cho riêng lớp Student dựa trên điểm

```
public int CompareTo(Object obj)
        {
            int iCompare = Convert.ToInt32(this.DiemTrungBinh - (obj as
Student).DiemTrungBinh);
           return iCompare;
        }
```

Bài tập 2: Tạo thêm 1 project ở trong cùng solution Lab01 có tên là "**Lab01-02**".

- ✓ Viết lại chương trình từ bài tập 1 trên theo cách tạo thêm một lớp là **Person** làm lớp cơ sở cho lớp **Student**. Chọn Họ tên làm field để đưa lên lớp **Person**.
- ✓ Thêm thông tin lớp giảng viên **Teacher** kế thừa từ lớp **Person.** Mỗi giảng viên đều có Mã số giảng viên, Họ Tên và Địa chỉ.
- ✓ Ở hàm **Main()** cho phép nhập vào tổng số N đối tượng được đưa vào **List** danh sách **Person**, ở mỗi lần nhập liệu user có quyền chọn là nhập thông tin cho **Student** hoặc **Teacher**. Xuất danh sách thông tin vừa nhập liệu trên.
- ✓ Sử dụng **Lambda expression** trong **thư viện LINQ** để thực hiện tìm kiếm danh sách các Sinh Viên có điểm trung bình cao nhất thuộc khoa "**CNTT**".

Hướng Dẫn Sử dụng Lambda expression trong LINQ

- Thêm namespace **System.Linq** vào lớp cần sử dụng

- Cú pháp của câu dạng truy vấn chỉ hỗ trợ một vài từ khóa đơn giản (Where, OrderBy, GroupBy, Take, Max, Min, Sum, Avarage, Count…)

- Ví dụ

```
//Lấy ra điểm TB cao nhất trong list SinhVien
   float maxScore = listStudent.Max(p => p.DiemTrungBinh);
//Lấy ra danh sách sinh viên có điểm bằng maxScore
   List<Student> listTemp = listStudent.Where(p => p.DiemTrungBinh == maxScore).ToList();

//Lấy DS Sinh Viên có điểm trung bình tăng dần
  List<Student> listSortStudent = listStudent.OrderBy(p => p.DiemTrungBinh).ToList();

//Lấy Sinh Viên có mã số = "CNTT020120"
  Student find1 = listStudent.FirstOrDefault(p => p.StudentID == "CNTT020120");
  Student find2 = listStudent.First(p => p.StudentID == "CNTT020120");

// Tìm kiếm danh sách các Sinh Viên có điểm trung bình cao nhất thuộc khoa "CNTT".
      //b1: lay ra list person ma là sinh vien khoa CNTT
   var listTempStudent = listPerson.Where(p => (p is Student) && (p as Student).Faculty ==
"CNTT").ToList();
     //b2: lây điêm cao nhất trong list tìmm kiếm
  var maxScore = listTempStudent.Max(p => (p as Student).AverageScore);
     //b3: lấy danh sách sinh viên
     var ListMaxStudent = listTempStudent.Where(p => (p as Student).AverageScore ==
maxScore);
```

✓ Viết phương thức tìm kiếm dữ liệu. Nhập vào một "Tên" để tìm kiếm, Xuất ra tất cả danh sách những sinh viên và giảng viên mà có tên chứa tên tìm kiếm không phân biệt các kí tự hoa thường.

```
//hướng dẫn: Lấy danh sách sinh viên có tên chứa findName không phân biệt hoa thường
     var listFindStudent = listPerson.Where(p => (p is Student) &&  (p as
    Student).FullName.ToLower().Contains(findName.ToLower())).ToList();
```

Bài tập 3: Thêm 1 project ở trong cùng solution Lab01 có tên là "**Lab01-03**".

Xây dựng chương trình quản lý danh sách hoá đơn tiền điện của khách hàng. Thông tin bao gồm các loại khách hàng:

- Khách hàng Việt Nam: mã khách hàng, họ tên, ngày ra hoá đơn (ngày, tháng, năm), đối tượng khách hàng (sinh hoạt, kinh doanh, sản xuất): số lượng (số KW tiêu thụ), đơn giá, định mức.
  Thành tiền được tính như sau:
  - Nếu số lượng <= định mức thì: thành tiền = số lượng * đơn giá.
  - Ngược lại thì: thành tiền = số lượng * đơn giá * định mức + số lượng KW vượt định mức * Đơn giá * 2.5.
- Khách hàng nước ngoài: mã khách hàng, họ tên, ngày ra hoá đơn (ngày, tháng, năm), quốc tịch, số lượng, đơn giá. Thành tiền được tính = số lượng * đơn giá.

Thực hiện các yêu cầu sau:

- ✓ Xây dựng các lớp với chức năng thừa kế
- ✓ Nhập xuất danh sách các hóa đơn khách hàng.
- ✓ Tính tổng số lượng cho từng loại khách hàng.
- ✓ Tính trung bình thành tiền của khách hàng người nước ngoài.
- ✓ Xuất ra các hoá đơn trong tháng 09 năm 2019 (cùa cả 2 loại khách hàng).
- ✓ Nhập vào một mã khách hàng để tìm kiếm. Xuất ra tổng thành tiền của mã số khách hàng vừa tìm kiếm (nếu có).

Bài tập 4. Thêm 1 project ở trong cùng solution Lab01 có tên là "**Lab01-04**".

Thư viên X quản lý danh sách các loại sách. Thông tin về các loại sách:

 Sách giáo khoa: Mã sách, ngày nhập (ngày, tháng, năm), đơn giá, số lượng, nhà xuất bản, tình trạng (mới, cũ).

Nếu tình trạng sách là mới thì: thành tiền = số lượng * đơn giá.

Nếu tình trạng sách là cũ thì: thành tiền = số lượng * đơn giá * 50%

 Sách tham khảo: Mã sách, ngày nhập (ngày, tháng, năm), đơn giá, số lượng, nhà xuất bản, thuế.

Thành tiền = số lượng * đơn giá + thuế

Thực hiện các yêu cầu sau:

- ✓ Xây dựng các lớp với chức năng thừa kế.
- ✓ Nhập xuất danh sách các loại sách.
- ✓ Tính tổng thành tiền cho từng loại.
- ✓ Tính trung bình cộng đơn giá của các sách tham khảo.
- ✓ Xuất ra các sách giáo khoa của nhà xuất bản "X".
- ✓ Nhập vào một mã sách để tìm kiếm. Xuất ra tổng thành tiền của mã sách vừa tìm kiếm (nếu có).

----------Hết Lab 01--------