

Lab 03

Arrays

Classes and Methods

Mục tiêu

- Khai báo, khởi tạo và thao tác với mảng
- Khai báo lớp và các thành viên của lớp
- Tạo đối tượng, truy cập các thành viên của lớp
- Làm việc với mảng đối tượng

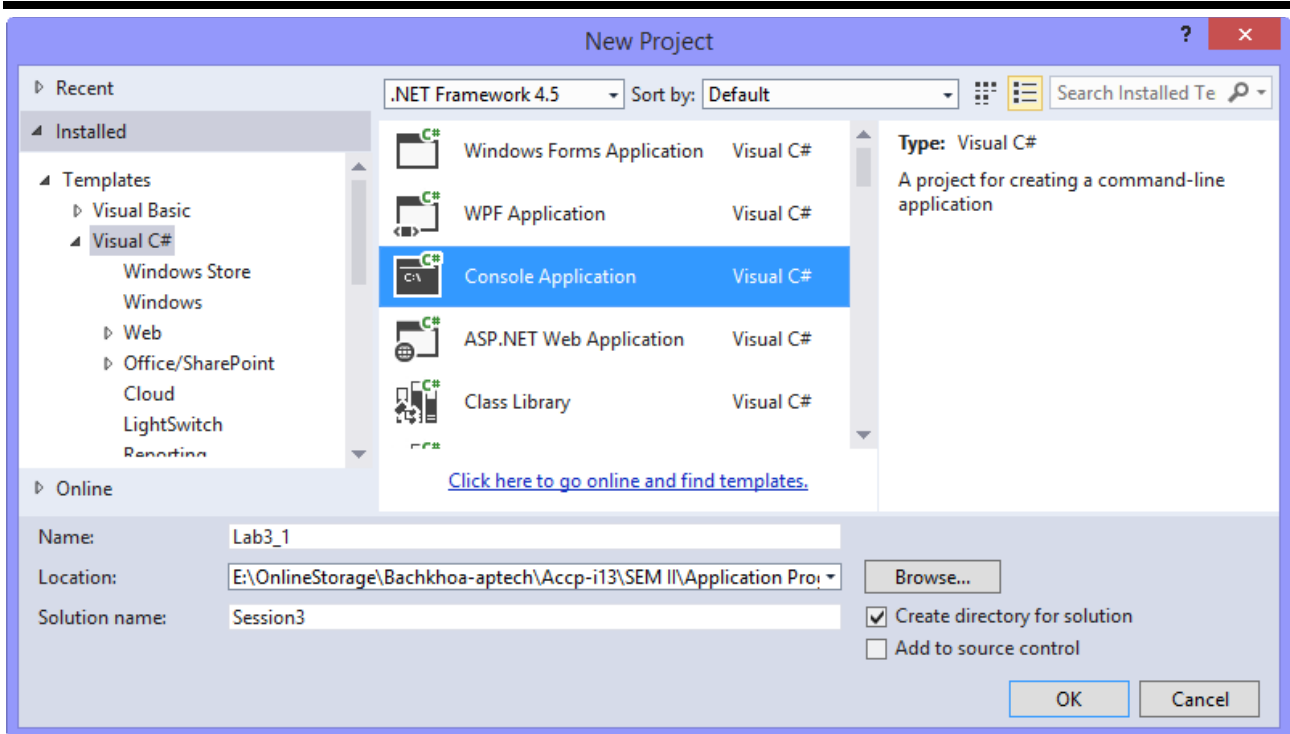
Phần I Bài tập step by step

Bài 3.1

Viết chương trình C# thực hiện một số thao tác trên mảng một chiều như sau:

- Khai báo và khởi tạo 1 mảng số nguyên khoảng 10 phần tử.
- In ra màn hình các phần tử của mảng.
- Tìm phần tử nhỏ nhất của mảng
- Kiểm tra xem mảng có đối xứng không.

Bước 1: Mở Visual Studio 2013, vào menu File -> New -> Project -> chọn loại project “Console Application”, nhập tên project, tên solution -> OK.

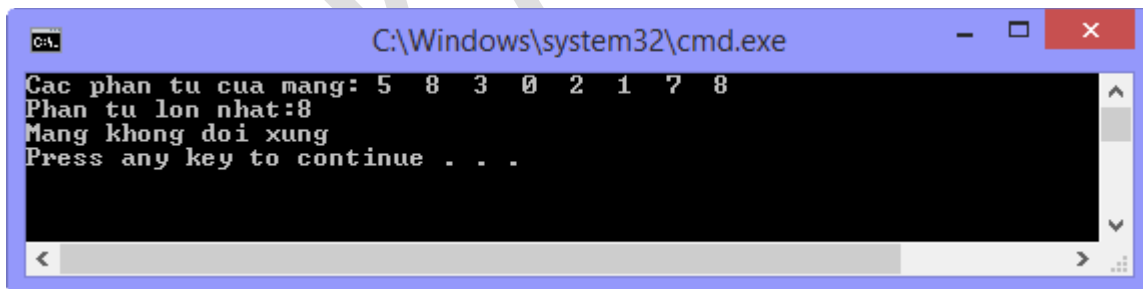


Bước 2: Mở tệp Program.cs và code cho hàm Main theo gợi ý sau:

```
/// <summary>
/// Một số thao tác với mảng 1 chiều
/// </summary>
/// <param name="args"></param>
static void Main(string[] args)
{
    //khai báo và khởi tạo mảng 1 chiều
    int[] m = { 5, 8, 3, 0, 2, 1, 7, 8 };
    //duyet mảng và in dữ liệu
    Console.Write("Cac phan tu cua mang:");
    for (int i = 0; i < m.Length; i++)
    {
        Console.Write(" {0} ", m[i]);
    }
    //tìm phần tử lớn nhất
    int max = m[0];
    for (int i = 0; i < m.Length; i++)
    {
        if (max < m[i])
```

```
        max = m[i];
    }
    Console.WriteLine("\nPhan tu lon nhat:" + max);
    //kiểm tra mảng có đối xứng không?
    bool kt = true;
    for (int i = 0; i < m.Length / 2; i++)
    {
        if (m[i] != m[m.Length - 1 - i])
        {
            kt = false;
            break;
        }
    }
    if (kt)
        Console.WriteLine("Mang doi xung");
    else
        Console.WriteLine("Mang khong doi xung");
}
```

Bước 3: Nhấn Ctrl+F5 để chạy và xem kết quả

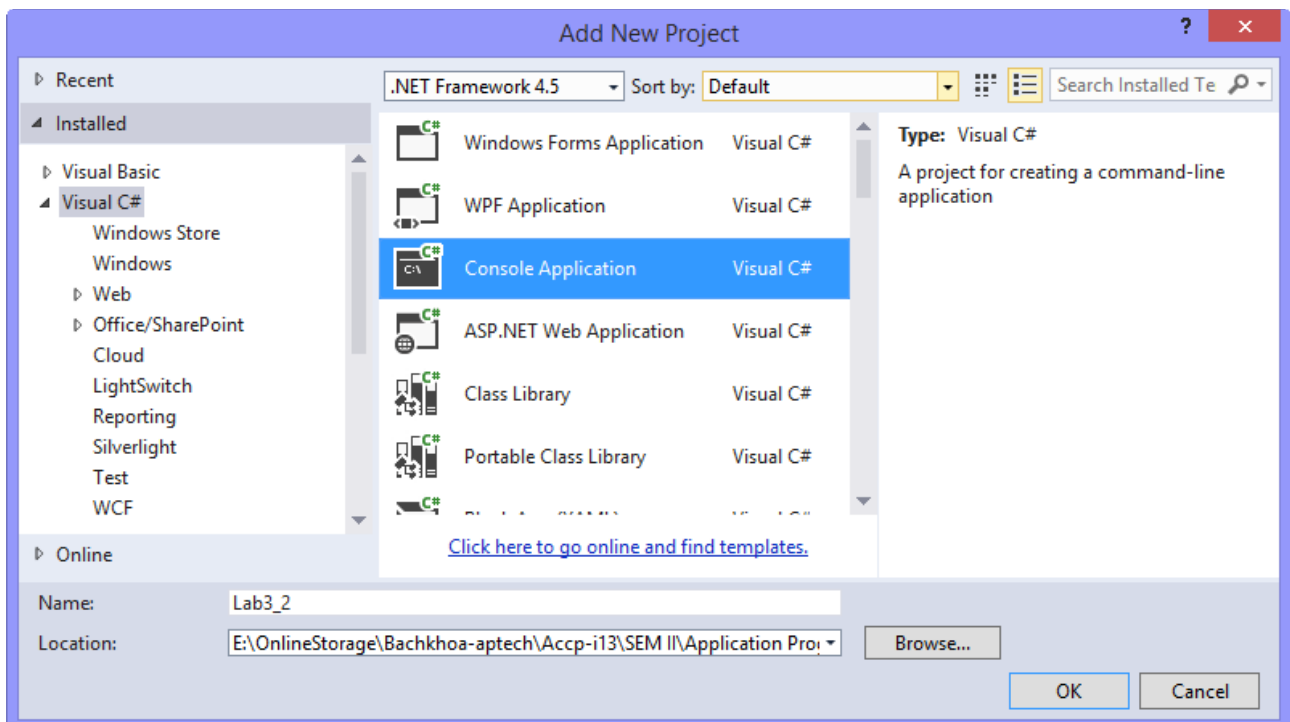


Bài 3.2

Viết chương trình C# thực hiện một số thao tác trên mảng hai chiều như sau:

- Khai báo vào khởi tạo mảng 2 chiều 4x3
- Duyệt mảng và in theo ma trận hàng cột
- In ra các phần tử có chỉ số hàng bằng chỉ số cột
- In ra các phần tử lớn nhất trên hàng

Bước 1: Kích chuột phải vào Solution “Session3” chọn Add -> New Project -> nhập tên.



Bước 2: Mở tệp Program.cs và code cho hàm Main theo gợi ý sau:

```
/// <summary>
/// Chương trình minh họa một số thao tác trên mảng 2 chiều
/// </summary>
/// <param name="args"></param>
static void Main(string[] args)
{
    //khai báo và khởi tạo mảng
    int[,] a ={
        {4,6,9},
        {2,4,5},
        {9,2,6},
        {1,6,3}
    };
    //duyet mảng và in theo hàng cột
    Console.WriteLine("Noi dung mang:");
    for (int i = 0; i <= a.GetUpperBound(0); i++)
    {
```

```
        Console.WriteLine();
        for (int j = 0; j <= a.GetUpperBound(1); j++)
        {
            Console.Write(" {0} ", a[i, j]);
        }
        Console.WriteLine();
    }

    //tìm các phần tử có chỉ số hàng bằng chỉ số cột
    Console.WriteLine("Cac phan tu co chi so hang bang chi so cot:");
    for (int i = 0; i <= a.GetUpperBound(0); i++)
    {
        for (int j = 0; j <= a.GetUpperBound(1); j++)
        {
            if (i == j)
                Console.Write(" {0} ", a[i, j]);
        }
    }

    //các phần tử lớn nhất trên hàng
    Console.WriteLine("Cac phan tu lon nhat tren hang");
    for (int i = 0; i <= a.GetUpperBound(0); i++)
    {
        int max = a[i, 0];
        for (int j = 0; j <= a.GetUpperBound(1); j++)
        {
            if (max < a[i, j])
                max = a[i, j];
        }
        Console.WriteLine("Hang {0}:{1}", i, max);
    }
}
```

Bước 3: Nhấn Ctrl+F5 để chạy và xem kết quả

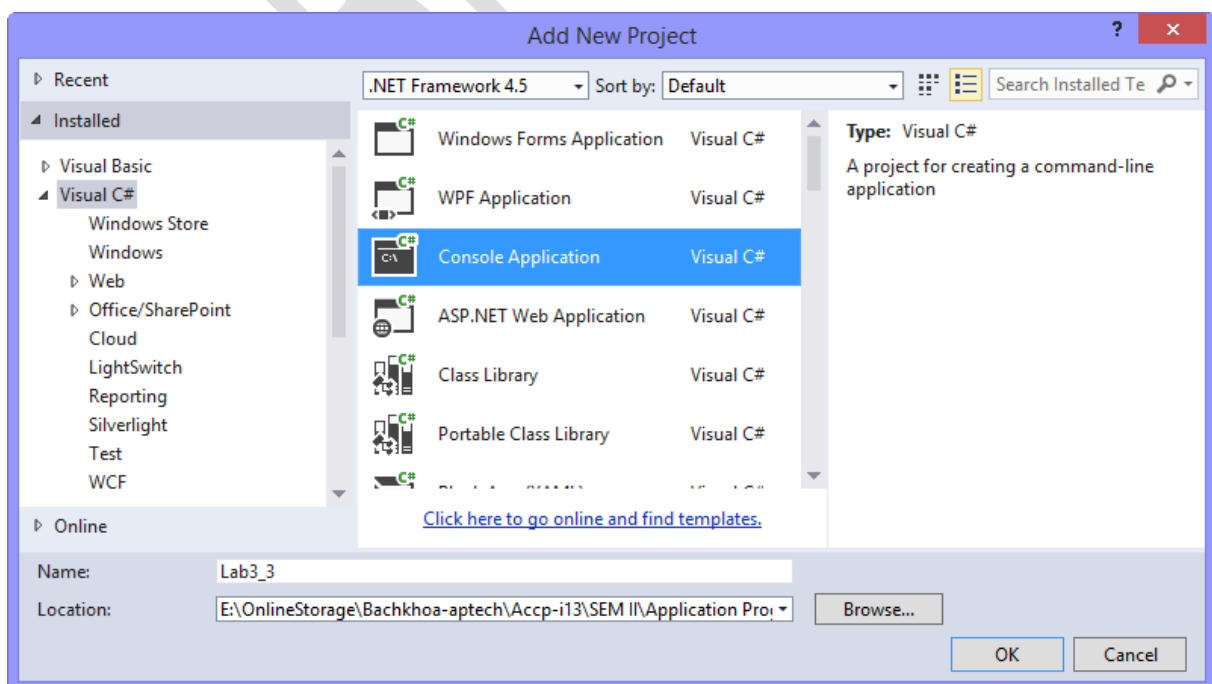


Bài 3.3

Viết chương trình C# thực hiện một số công việc sau:

- Tạo lớp Car gồm các trường (make, model, color, year) có phạm vi là public
- Tạo phương thức Start và Stop in ra thông báo về tình trạng của xe.
- Trong hàm Main khởi tạo đối tượng Car, gán giá trị cho các trường, hiển thị thông tin xe, gọi phương thức Start, Stop.

Bước 1: Kích chuột phải vào Solution “Session3” chọn Add -> New Project ->nhập tên.



Bước 2: Kích chuột phải vào project “Lab3_3” chọn Add -> Class -> đặt tên là “Car” sau đó code theo gợi ý sau:

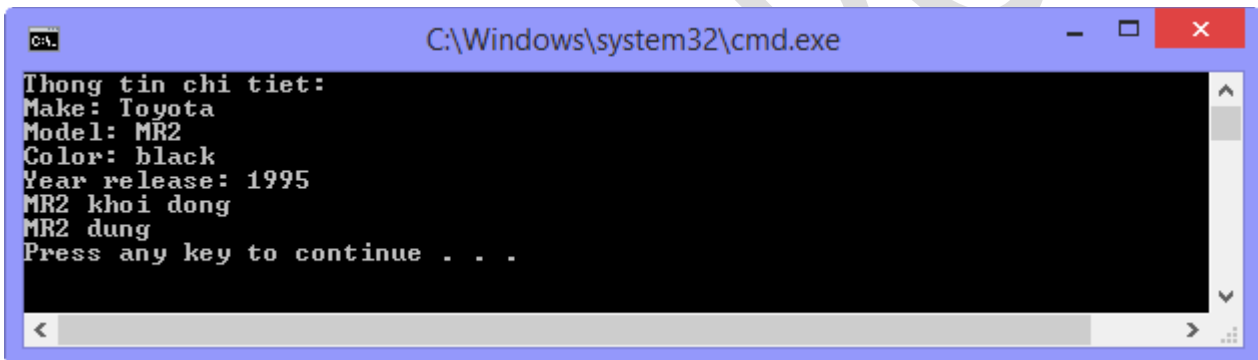
```
class Car
{
    // khai báo các trường với phạm vi public
    public string make;
    public string model;
    public string color;
    public int year;
    // định nghĩa phương thức start
    public void Start()
    {
        System.Console.WriteLine(model + " khoi dong");
    }
    // định nghĩa phương thức stop
    public void Stop()
    {
        System.Console.WriteLine(model + " dung");
    }
}
```

Bước 3: Mở tệp Program.cs và code cho hàm Main theo gợi ý sau:

```
/// <summary>
/// Minh họa việc khởi tạo đối tượng, truy cập các trường và phương thức
/// </summary>
/// <param name="args"></param>
static void Main(string[] args)
{
    // tạo đối tượng Car
    Car myCar = new Car();
    // gán giá trị cho các trường của đối tượng
    myCar.make = "Toyota";
    myCar.model = "MR2";
    myCar.color = "black";
    myCar.year = 1995;
```

```
//hiển thị thông tin về đối tượng myCar
System.Console.WriteLine("Thông tin chi tiết:");
System.Console.WriteLine("Make: " + myCar.make);
System.Console.WriteLine("Model: " + myCar.model);
System.Console.WriteLine("Color: " + myCar.color);
System.Console.WriteLine("Year release: " + myCar.year);
// gọi các phương thức
myCar.Start();
myCar.Stop();
}
```

Bước 4: Nhấn Ctrl+F5 để chạy và xem kết quả

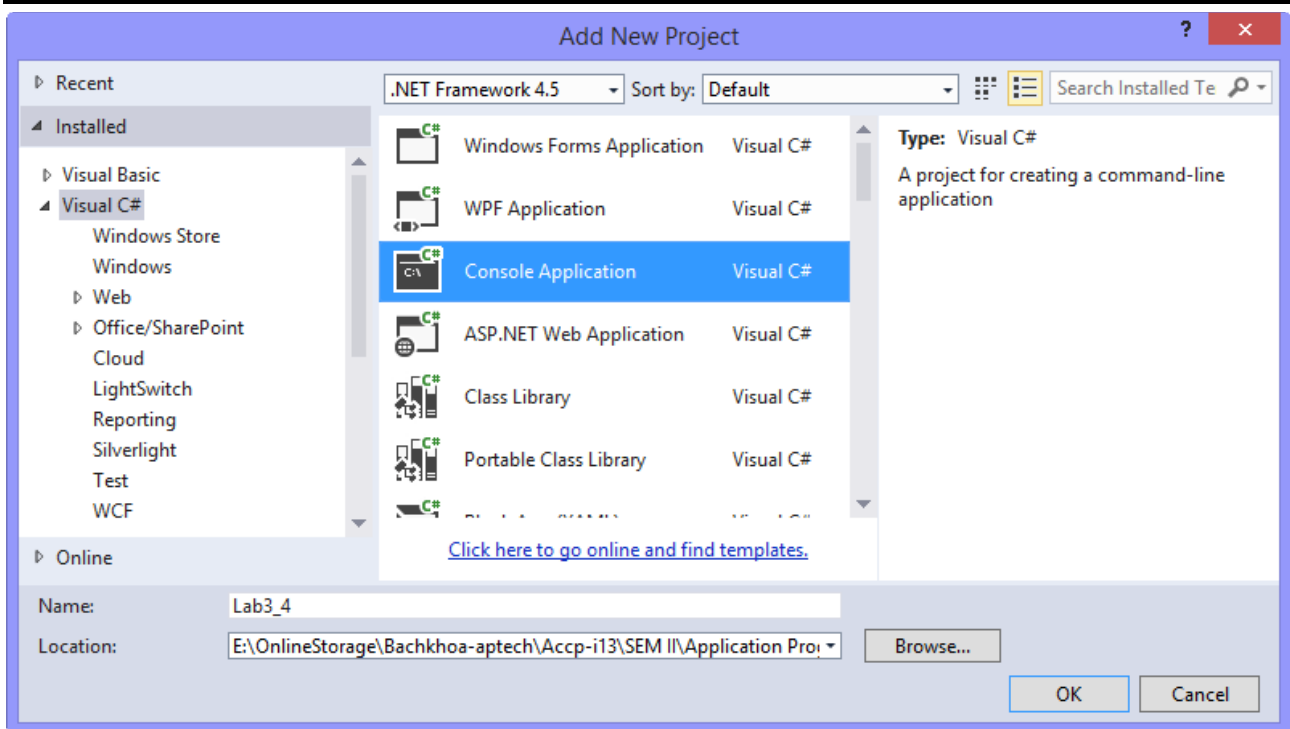


Bài 3.4

Viết chương trình C# thực hiện một số công việc sau:

- Tạo lớp Contact với các trường (id, firstname, lastname, address, phone, email), các trường khai báo với phạm vi private.
- Tạo các thuộc tính cho các trường.
- Viết phương thức hiển thị thông tin contact.
- Tạo 2 đối tượng và gọi phương thức để test.

Bước 1: Kích chuột phải vào Solution “Session3” chọn Add -> New Project ->nhập tên.



Bước 2: Kích chuột phải vào project “Lab3_4” chọn Add -> Class -> đặt tên là “Contact” sau đó code theo gợi ý sau:

```
class Contact
{
    //khai báo các trường
    private int id;
    private string firstname;
    private string lastname;
    private string address;
    private string phone;
    private string email;
    //tạo constructor không tham số
    public Contact()
    {
    }
    //tạo constructor với các tham số
    public Contact(int id, string firstname, string lastname,
        string address, string phone, string email)
    {
        this.id = id;
    }
}
```

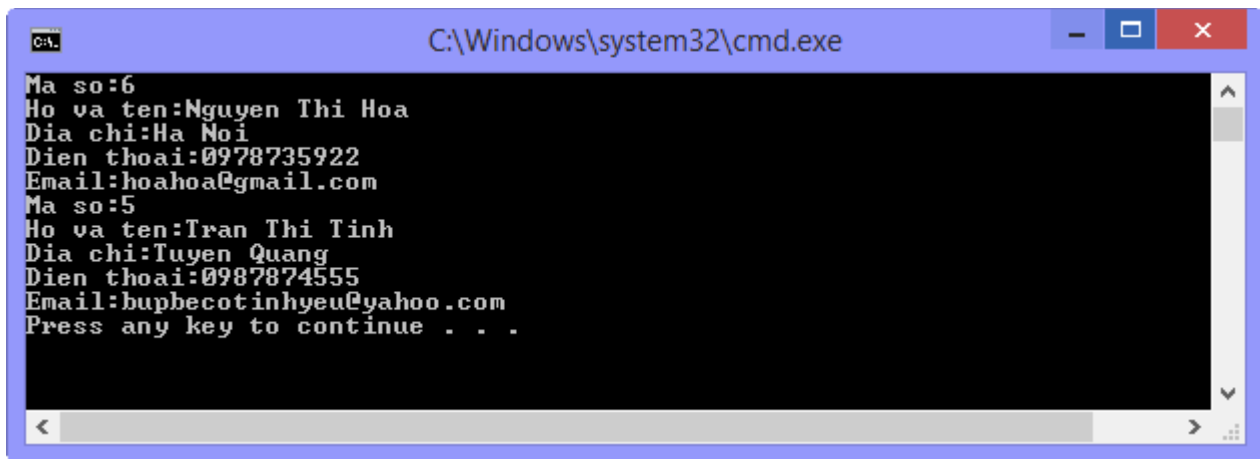
```
this.firstname = firstname;
this.lastname = lastname;
this.address = address;
this.phone = phone;
this.email = email;
}
//tạo thuộc tính cho các trường
public int Id
{
    get { return id; }
    set { id = value; }
}
public string FirstName
{
    get { return firstname; }
    set { firstname = value; }
}
public string LastName
{
    get { return lastname; }
    set { lastname = value; }
}
public string Address
{
    get { return address; }
    set { address = value; }
}
public string Phone
{
    get { return phone; }
    set { phone = value; }
}
public string Email
{
    get { return email; }
    set { email = value; }
```

```
}  
//định nghĩa phương thức display  
public void Display()  
{  
    Console.WriteLine("Ma so:" + id);  
    Console.WriteLine("Ho va ten:{0} {1}", firstname, lastname);  
    Console.WriteLine("Dia chi:" + address);  
    Console.WriteLine("Dien thoai:" + phone);  
    Console.WriteLine("Email:" + email);  
}  
}
```

Bước 3: Mở tệp Program.cs và code cho hàm Main theo gợi ý sau:

```
static void Main(string[] args)  
{  
    //tạo đối tượng contact với constructor không tham số  
    Contact ct1 = new Contact();  
    //gán giá trị cho ct1  
    ct1.Id = 6;  
    ct1.FirstName = "Nguyen Thi";  
    ct1.LastName = "Hoa";  
    ct1.Address = "Ha Noi";  
    ct1.Phone = "0978735922";  
    ct1.Email = "hoahoa@gmail.com";  
    //tạo đối tượng contact với constructor có tham số  
    Contact ct2 = new Contact(5, "Tran Thi", "Tinh", "Tuyen Quang",  
        "0987874555", "bupbecotinhyeu@yahoo.com");  
    //hiển thị thông tin ct1 và ct2  
    ct1.Display();  
    ct2.Display();  
}
```

Bước 4: Nhấn Ctrl+F5 để chạy và xem kết quả



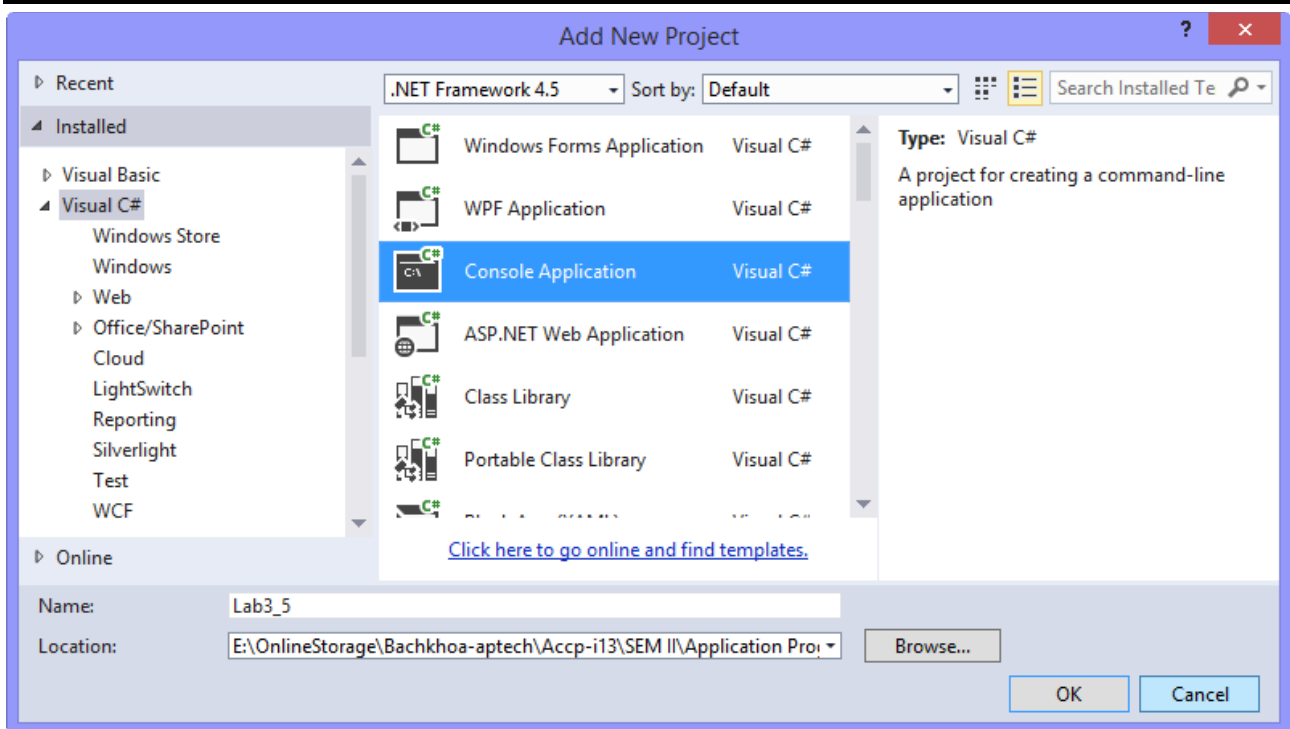
```
C:\Windows\system32\cmd.exe
Ma so:6
Ho va ten:Nguyen Thi Hoa
Dia chi:Ha Noi
Dien thoai:0978735922
Email:hoahoa@gmail.com
Ma so:5
Ho va ten:Tran Thi Tinh
Dia chi:Tuyen Quang
Dien thoai:0987874555
Email:bupbecotinhyeu@yahoo.com
Press any key to continue . . .
```

Bài 3.5

Viết chương trình C# thực hiện một số công việc sau:

- Tạo lớp Student với các thuộc tính khai báo tự động (Id, Name, Age).
- Tạo phương thức Display để hiển thị thông tin sinh viên.
- Tạo lớp StudentModel.cs
- Khởi tạo một danh sách gồm 5 sinh viên trong StudentModel.
- Tạo các phương thức overloading “GetStudent” trong lớp StudentModel để:
 - o Trả về toàn bộ danh sách sinh viên
 - o Trả về sinh viên theo Id.
 - o Trả về sinh viên có tuổi từ x->y.
- Khởi tạo StudentModel và test trong Main.

Bước 1: Kích chuột phải vào Solution “Session3” chọn Add -> New Project -> nhập tên.



Bước 2: Kích chuột phải vào project “Lab3_5” chọn Add -> Class -> đặt tên là “Student” sau đó code theo gợi ý sau:

```
class Student
{
    //tạo các thuộc tính tự động
    public int Id { get; set; }
    public string Name { get; set; }
    public int Age { get; set; }
    //phương thức hiển thị
    public void Display()
    {
        Console.WriteLine("Id:" + Id);
        Console.WriteLine("Name:" + Name);
        Console.WriteLine("Age:" + Age);
    }
}
```

Bước 3: Kích chuột phải vào project “Lab3_5” chọn Add -> Class -> đặt tên là “Student” sau đó code theo gợi ý sau:

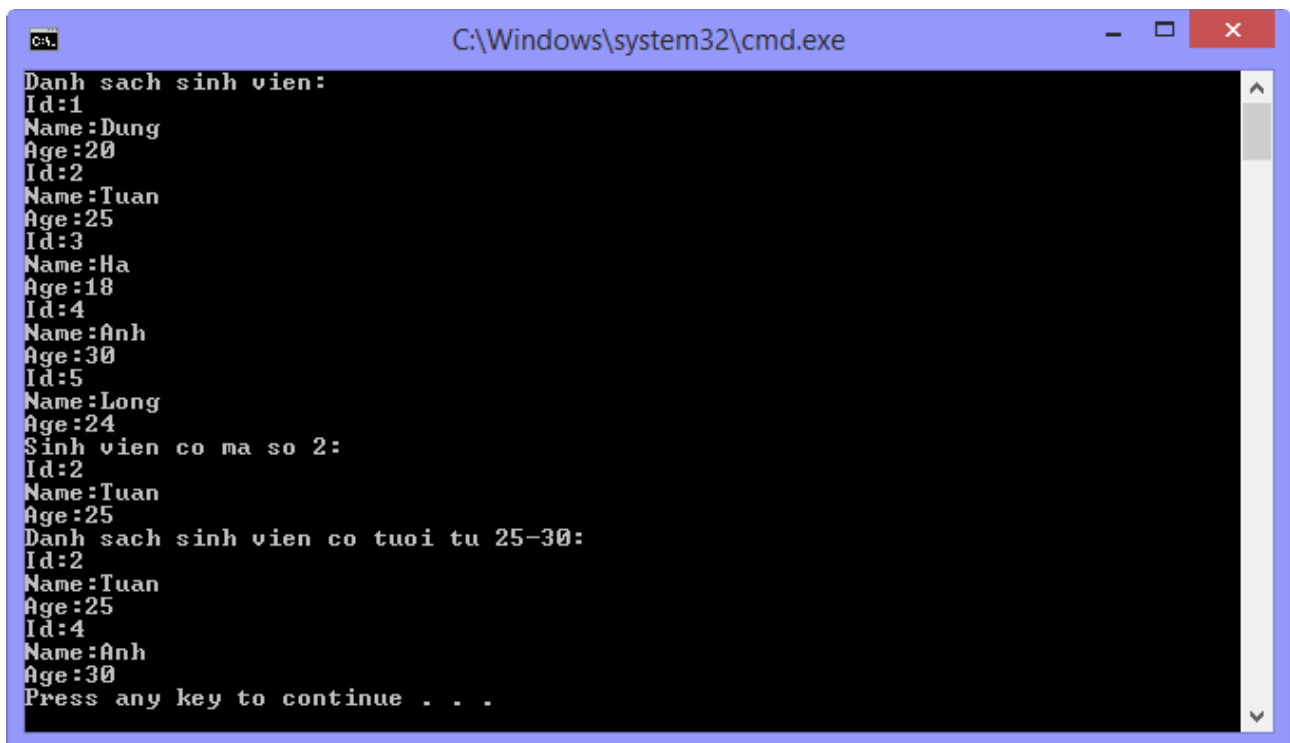
```
class StudentModel
{
    //khai báo danh tập hợp chứa các sinh viên
    List<Student> liststudent;
    //phương thức khởi tạo
    public StudentModel()
    {
        //khởi tạo danh sách sinh viên
        this.liststudent = new List<Student> {
            new Student(){Id=1,Name="Dung",Age=20},
            new Student(){Id=2,Name="Tuan",Age=25},
            new Student(){Id=3,Name="Ha",Age=18},
            new Student(){Id=4,Name="Anh",Age=30},
            new Student(){Id=5,Name="Long",Age=24},
        };
    }
    #region các phương thức overloading
    //phương thức trả về danh sách sinh viên
    public List<Student> GetStudent()
    {
        return liststudent;
    }
    //phương thức trả về sinh viên theo id
    public Student GetStudent(int id)
    {
        Student st=null;
        foreach (var item in liststudent)
        {
            if (item.Id == id)
                st = item;
        }
        return st;
    }
    //phương thức trả về sinh viên có tuổi từ x to y
    public List<Student> GetStudent(int x, int y)
    {
```

```
List<Student> result = new List<Student>();  
foreach (var item in liststudent)  
{  
    if (item.Age >= x && item.Age <= y)  
        result.Add(item);  
}  
return result;  
}  
#endregion  
}
```

Bước 4: Mở tệp Program.cs và code cho hàm Main theo gợi ý sau:

```
static void Main(string[] args)  
{  
    //tạo đối tượng  
    StudentModel action = new StudentModel();  
    //gọi phương thức lấy tất cả sinh viên  
    List<Student> listall = action.GetStudent();  
    //hiển thị  
    foreach (var item in listall)  
    {  
        item.Display();  
    }  
    //gọi phương thức lấy sinh viên theo id  
    Student st = action.GetStudent(2);  
    //hiển thị  
    st.Display();  
    //gọi phương thức lấy sinh viên có tuổi từ 25-30  
    List<Student> listage = action.GetStudent(25, 30);  
    //hiển thị  
    foreach (var item in listage)  
    {  
        item.Display();  
    }  
}
```

Bước 5: Nhấn Ctrl+F5 để chạy và xem kết quả

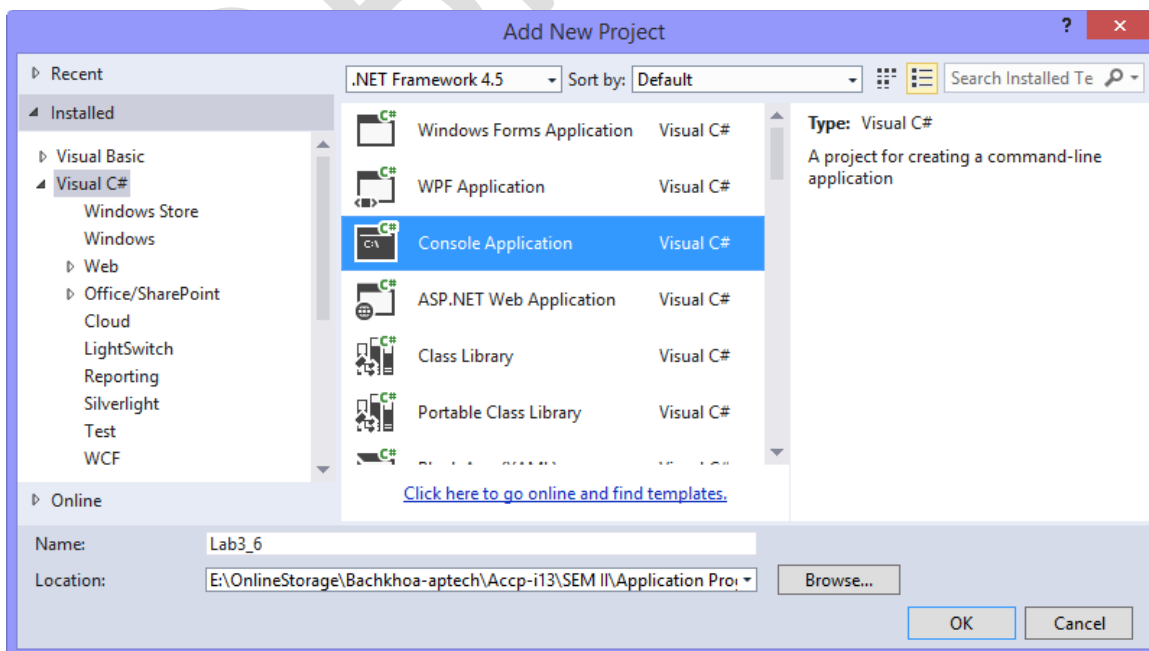


```
C:\Windows\system32\cmd.exe
Danh sach sinh vien:
Id:1
Name:Dung
Age:20
Id:2
Name:Tuan
Age:25
Id:3
Name:Ha
Age:18
Id:4
Name:Anh
Age:30
Id:5
Name:Long
Age:24
Sinh vien co ma so 2:
Id:2
Name:Tuan
Age:25
Danh sach sinh vien co tuoi tu 25-30:
Id:2
Name:Tuan
Age:25
Id:4
Name:Anh
Age:30
Press any key to continue . . .
```

Bài 3.6

Viết chương trình C# minh họa việc sử dụng từ khóa ref, out:

Bước 1: Kích chuột phải vào Solution “Session3” chọn Add -> New Project -> nhập tên.



Bước 2: Kích chuột phải vào project “Lab3_6” chọn Add -> Class -> đặt tên là “Utility” sau đó code theo gợi ý sau:

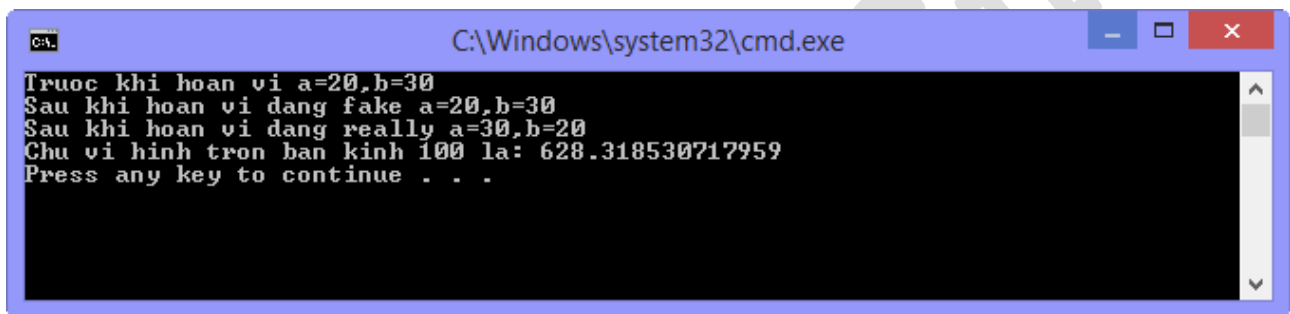
```
class Utility
{
    //Không sử dụng ref
    public void SwapFake(int a,int b)
    {
        int tg = a;
        a = b;
        b = tg;
    }
    //Sử dụng ref cho tham số
    public void SwapReally(ref int a, ref int b)
    {
        int tg = a;
        a = b;
        b = tg;
    }
    //Sử dụng out cho tham số
    public void AreaCircle(double radius, out double area)
    {
        area = 2 * Math.PI * radius;
    }
}
```

Bước 3: Mở tệp Program.cs và code cho hàm Main theo gợi ý sau:

```
/// <summary>
/// Chương trình minh họa việc sử dụng từ khóa ref và out
/// </summary>
/// <param name="args"></param>
static void Main(string[] args)
{
    int a = 20, b = 30;
    double area;
    Utility u = new Utility();
```

```
Console.WriteLine("Truoc khi hoan vi a={0},b={1}", a, b);  
u.SwapFake(a, b);  
Console.WriteLine("Sau khi hoan vi dang fake a={0},b={1}", a, b);  
u.SwapReally(ref a, ref b);  
Console.WriteLine("Sau khi hoan vi dang really a={0},b={1}", a, b);  
u.AreaCircle(100, out area);  
Console.WriteLine("Chu vi hinh tron ban kinh 100 la: {0}", area);  
}
```

Bước 5: Nhấn Ctrl+F5 để chạy và xem kết quả



Phần II Bài tập tự làm

Bài 3.1: Tạo ứng dụng C# và thực hiện những công việc sau.

- Khởi tạo một mảng 1 chiều số nguyên gồm 10 phần tử.
- Tìm phần tử nhỏ nhất của mảng.
- Đảo ngược mảng.
- Sắp xếp mảng tăng hoặc giảm.
- Tìm phần tử là số nguyên tố.
- In ra số lượng các số dương liên tiếp nhiều nhất.
- Tính trung bình cộng các phần tử dương.
- Kiểm tra xem mảng có phải chứa các phần tử âm dương, đan xen nhau không?

Bài 3.2: Tạo ứng dụng C# và thực hiện những công việc sau.

- Khai báo và khởi tạo một mảng 2 chiều với 4 dòng và 4 cột.

- In mảng theo hàng cột.
- Tính tổng các phần tử mà có chỉ số hàng=chỉ số cột.
- In ra các phần tử nhỏ nhất trên cột.
- Liệt kê các phần tử chia hết cho 7.
- Tìm tổng các phần tử nằm trên đường viền của mảng.
- Chuyển thành mảng 1 chiều rồi sắp xếp tăng dần.

Bài 3.3: Tạo ứng dụng C# và thực hiện công việc theo mô tả sau:

Cho 2 mảng như sau.

Mảng CAN 10 phần tử

Giáp	Ất	Bính	Đinh	Mậu	Kỷ	Canh	Tân	Nhâm	Quý
------	----	------	------	-----	----	------	-----	------	-----

Mảng CHI 12 phần tử

Tí	Sửu	Dần	Mão	Thìn	Tỵ	Ngọ	Mùi	Thân	Dậu	Tuất	Hợi
----	-----	-----	-----	------	----	-----	-----	------	-----	------	-----

- Theo lịch âm thì mỗi năm sẽ ứng với CAN+CHI.
- Giả sử năm 2010 là năm CANH DẦN.
- Hãy nhập vào 1 năm bất kỳ và in ra năm âm tương ứng, bạn có thể đổi lại vị trí bắt đầu của CAN CHI cho phù hợp.

Bài 3.4: Viết ứng dụng C# thực hiện những công việc sau:

Tạo lớp Employee (nhân viên) với mô tả:

- Các thành phần dữ liệu
 - o **id:** Định danh, kiểu int.
 - o **name:** Họ tên nhân viên, kiểu String.
 - o **yearOfBirth:** Năm sinh nhân viên, kiểu int.
 - o **salaryLevel:** Bậc lương, kiểu double.
 - o **basicSalary:** Lương cơ bản kiểu double. (Chú ý lương cơ bản là trường được sử dụng chung cho mọi đối tượng của lớp Employee).

- **GetIncome()**: trả lại thu nhập của nhân viên. Thu nhập được tính bằng bậc lương nhân lương cơ bản ($\text{salaryLevel} * \text{basicSalary}$).
- **Display()**: hiển thị thông tin về nhân viên. Bao gồm các thông tin: định danh, tên, năm sinh, lương cơ bản, thu nhập.
- Hãy viết chương trình cài đặt lớp Employee và lớp sử dụng Employee.

Bài 3.5: Tạo lớp Book để biểu diễn quyển sách

- ✓ Các trường.
 - string author (tác giả).
 - int pages (số trang).
 - string isbn (mã số tiêu chuẩn quốc tế).
 - string title (tiêu đề).
 - int current page (trang hiện tại).
- ✓ Tạo constructor mặc định và thiết lập currentpage cho 1.
- ✓ Tạo constructor với tất cả các tham số để khởi tạo các trường tương ứng.
- ✓ Tạo phương thức flipPageForward (lật sang trang trước).
- ✓ Tạo phương thức flipPageBackward (lật sang trang sau).
- ✓ Viết chương trình để kiểm thử tất các tính năng của Book.

HẾT