

Lab 09

Advanced Methods and Types

Mục tiêu	2
Phần I Bài tập step by step	3
Bài 9.1	3
Bài 9.2	6
Bài 9.3	8
Bài 9.4	10
Bài 9.5	13
Phần II Bài tập tự làm	15
Bài 9.1	15
Bài 9.2	15

Mục tiêu

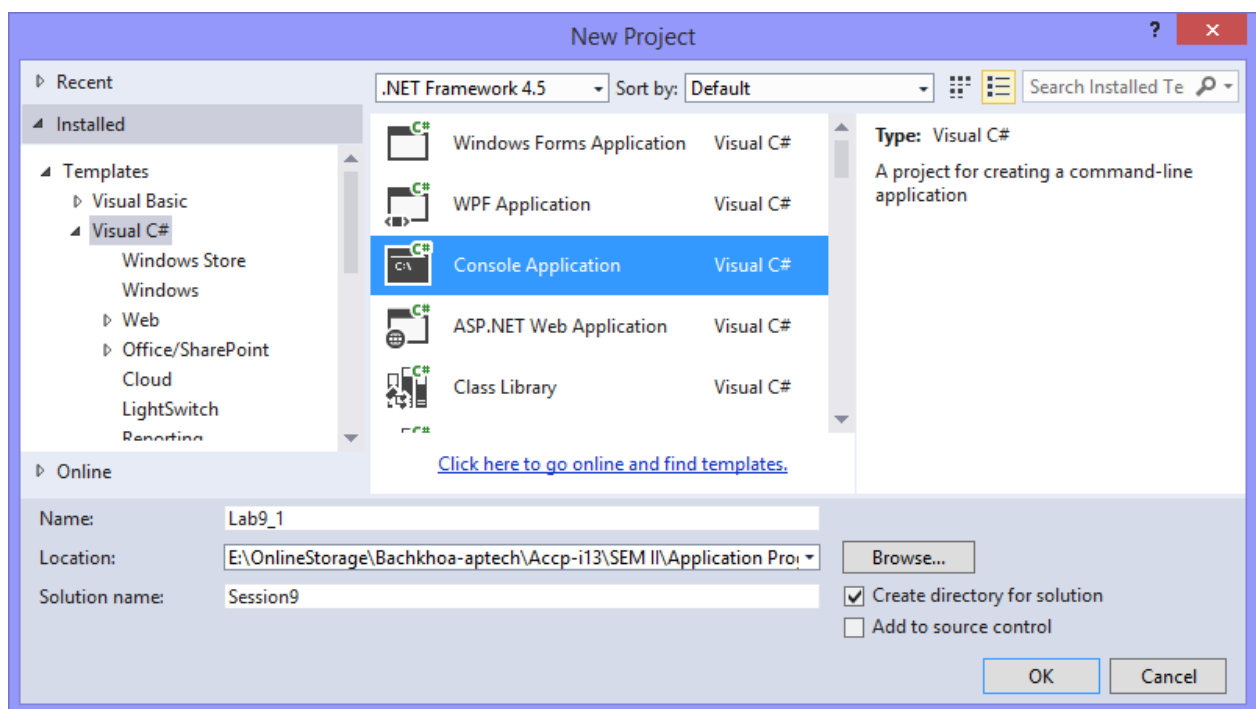
- Tạo và sử dụng phương thức mặc định
- Định nghĩa phương thức mở rộng
- Tạo và sử dụng kiểu mặc định
- Tạo và sử dụng kiểu thành phần
- Sử dụng các kiểu có khả năng chứa giá trị null

Phần I Bài tập step by step

Bài 9.1

Viết chương trình C# minh họa việc tạo delegate và gọi một phương thức nặc danh, nhiều phương thức nặc danh.

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



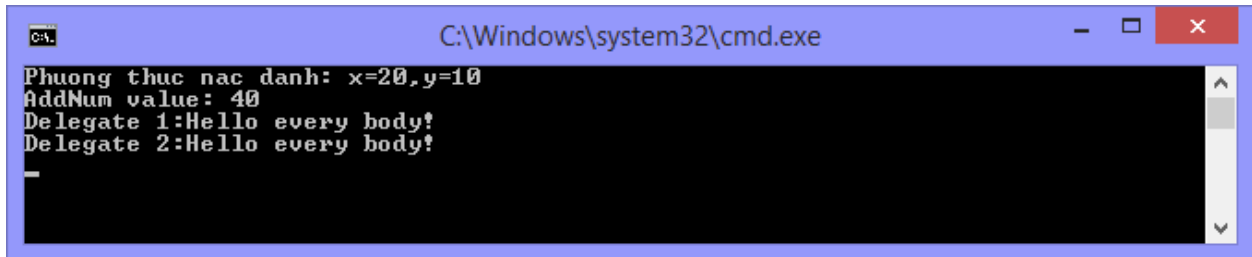
BƯỚC 2: Mở tệp Program.cs và code theo gợi ý sau:

```
//khai báo Delegate tính toán 2 số nguyên
delegate void Calculator(int num1, int num2);
//khai báo delegate hiển thị 1 thông báo
delegate void ShowMessage(string msg);
class Program
{
    //khai báo biến tĩnh
    static int num = 0;
```

```
//Định nghĩa phương thức tính cộng 2 số
public static void AddNum(int p, int q)
{
    num = p + q;
    Console.WriteLine("AddNum value: {0}", num);
}
/// <summary>
/// Phương thức Main
/// </summary>
/// <param name="args"></param>
static void Main(string[] args)
{
    //Tạo thẻ hiện của delegate sử dụng phương thức nặc danh
    Calculator pd = delegate(int x, int y)
    {
        Console.WriteLine("Phuong thuc nac danh: x={0},y={1}", x, y);
    };
    //gọi delegate sử dụng phương thức nặc danh
    pd(20, 10);
    //Tạo mới delegate sử dụng phương thức tường minh
    pd = new Calculator(AddNum);
    //gọi delegate sử dụng phương thức tường minh
    pd(10, 30);
    //Tạo thẻ hiện của delegate sử dụng phương thức nặc danh
    ShowMessage sm = delegate(string msg)
    {
        Console.WriteLine("Delegate 1:" + msg);
    };
    //Tham chiếu tới nhiều phương thức nặc danh
    sm += delegate(string msg)
    {
        Console.WriteLine("Delegate 2:" + msg);
    };
    //Gọi delegate sử dụng nhiều phương thức nặc danh
    sm("Hello every body!");
}
```

```
        Console.Read();  
    }  
}
```

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



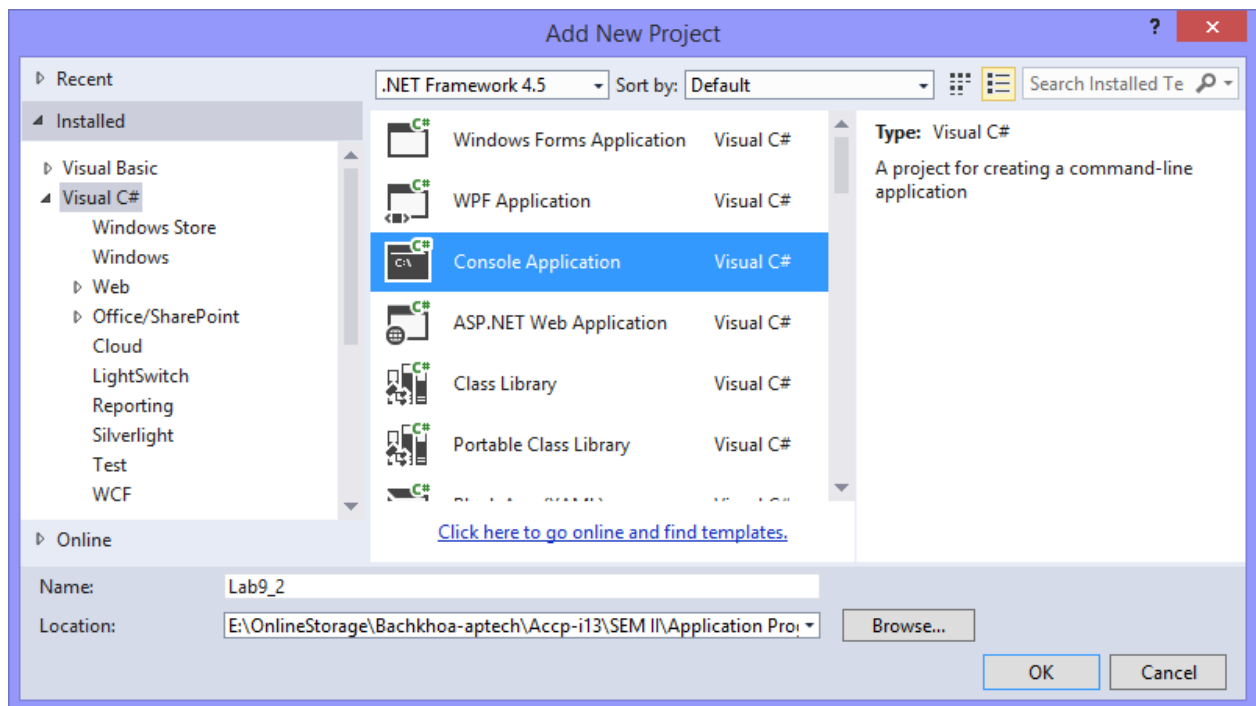
The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is displayed as follows:

```
Phuong thuc nac danh: x=20,y=10  
AddNum value: 40  
Delegate 1:Hello every body!  
Delegate 2:Hello every body!  
_
```

Bài 9.2

Viết chương trình C# minh họa việc tạo phương thức mở rộng cho lớp String để đếm số từ trong 1 chuỗi.

BƯỚC 1: Kích chuột phải vào Solution “Session9” chọn Add -> New Project ->nhập tên.



BƯỚC 2: Tạo lớp StringExtension theo code gợi ý sau:

```
//Định nghĩa lớp tĩnh
public static class StringExtension
{
    //định nghĩa phương thức mở rộng cho lớp String để đếm số từ của một chuỗi
    public static int WordCount(this String str)
    {
        return str.Split(new char[] { ' ', '.', '?' },
            StringSplitOptions.RemoveEmptyEntries).Length;
    }
}
```

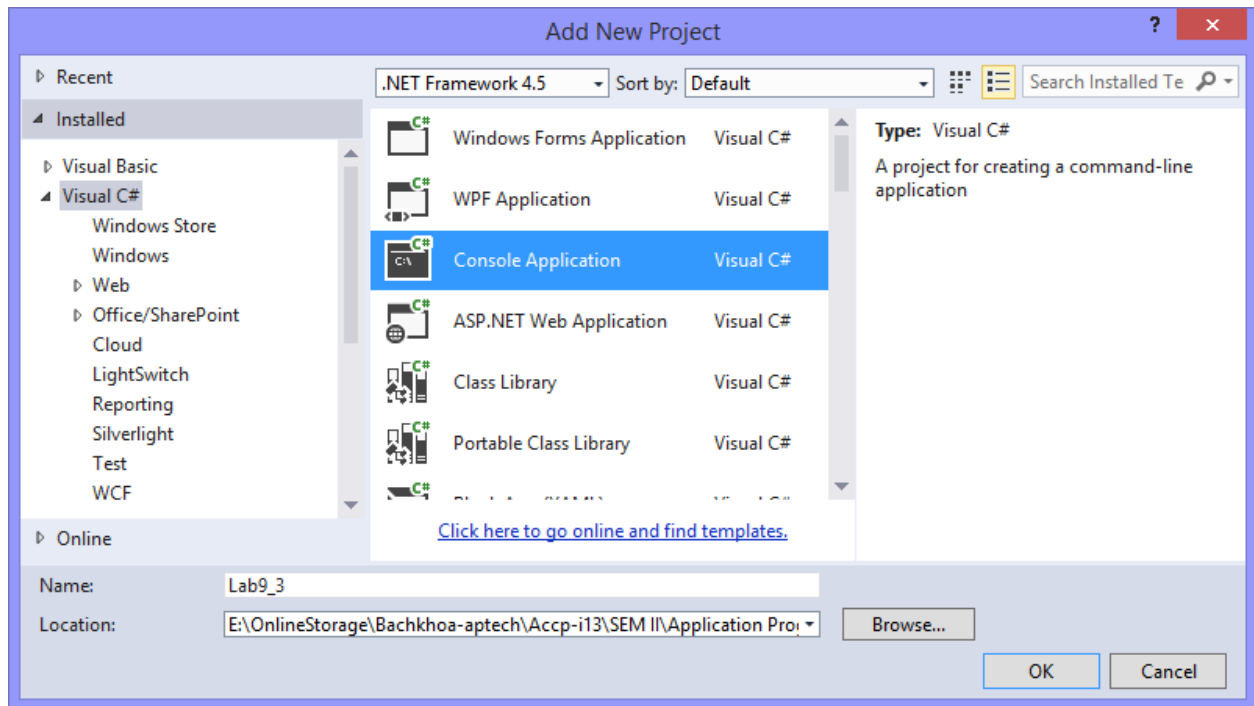
Bước 3: Mở tệp Program.cs và code theo gợi ý sau:

```
class Program
{
    static void Main(string[] args)
    {
        string s = "Ban that la tuyet voi.";
        //gọi phương thức mở rộng
        int count = s.WordCount();
        System.Console.WriteLine("Tong so tu cua s la: {0}", count);
    }
}
```

Bài 9.3

Viết chương trình C# minh họa việc tạo đối tượng nặc danh.

BƯỚC 1: Kích chuột phải vào Solution “Session9” chọn Add -> New Project ->nhập tên.



BƯỚC 2: Mở tệp Program.cs và code theo gợi ý sau:

```
class Program
{
    static void Main(string[] args)
    {
        //Tạo đối tượng nặc danh
        var employee = new
        {
            Name = "Lai Duc Chung",
            Age = 36,
            Address = "Ha Nam",
            Email = "chungld@bachkhoa-aptech.com",
            Married = 1
        };
    }
}
```



```
//In thông tin đối tượng nặc danh
Console.WriteLine("Ho va ten: " + employee.Name);
Console.WriteLine("Tuoi: " + employee.Age);
Console.WriteLine("Dia chi: " + employee.Address);
Console.WriteLine("Email: " + employee.Email);
Console.WriteLine("Tinh trang: " + employee.Married);
    }
}
```

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

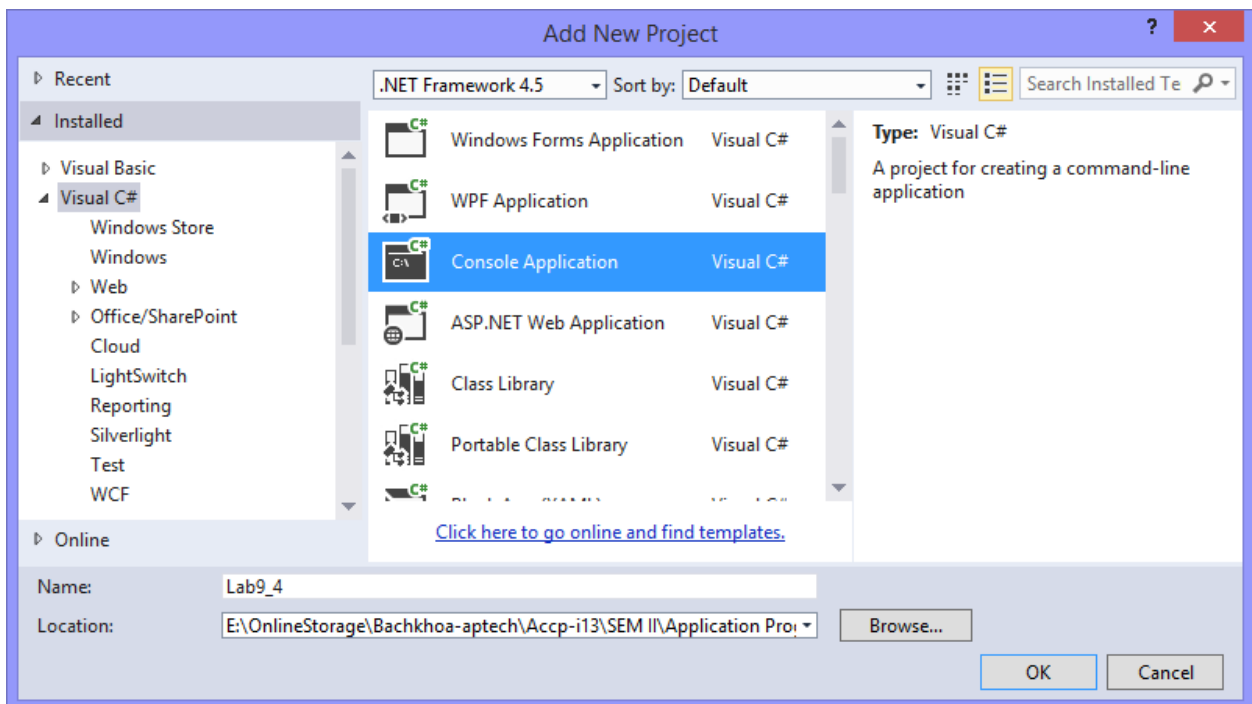


```
C:\Windows\system32\cmd.exe
Ho va ten: Lai Duc Chung
Tuoi: 36
Dia chi: Ha Nam
Email: chungld@bachkhoa-aptech.com
Tinh trang: 1
Press any key to continue . . . _
```

Bài 9.4

Viết chương trình C# minh họa việc tạo lớp thành phần, phương thức thành phần và sử dụng chúng.

BƯỚC 1: Kích chuột phải vào Solution “Session9” chọn Add -> New Project ->nhập tên.



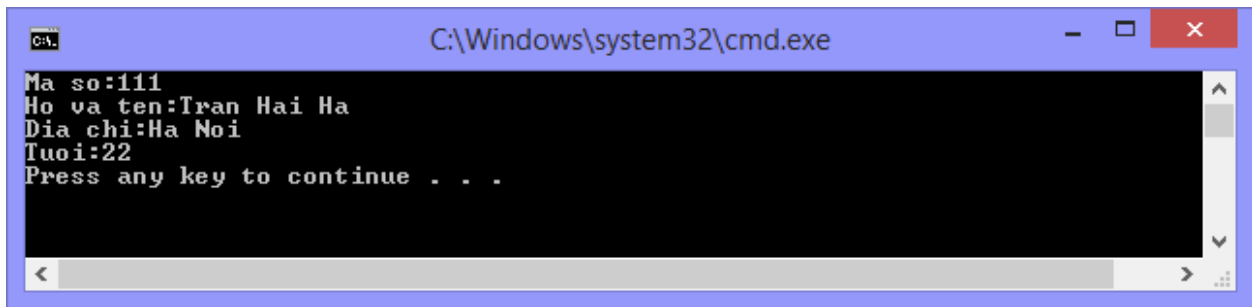
BƯỚC 2: Tạo lớp thành phần Employee:

```
//Định nghĩa lớp thành phần Employee
partial class Employee
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Address { get; set; }
    public int Age { get; set; }
    //định nghĩa phương thức thành phần Display
    partial void Display();
}
```

BƯỚC 3: Mở tệp Program.cs và code theo gợi ý sau:

```
//Định nghĩa lớp thành phần Employee
partial class Employee
{
    //constructor khởi tạo
    public Employee()
    {
        this.Name = "Tran Hai Ha";
        this.Address = "Ha Noi";
        this.Id = 111;
        this.Age = 22;
    }
    //triển khai phương thức thành phần
    partial void Display()
    {
        Console.WriteLine("Ma so:" + Id);
        Console.WriteLine("Ho va ten:" + Name);
        Console.WriteLine("Dia chi:" + Address);
        Console.WriteLine("Tuoi:" + Age);
    }
    //Định nghĩa phương thức gọi phương thức thành phần
    public void CallDisplay()
    {
        Display();
    }
}
class Program
{
    static void Main(string[] args)
    {
        //tạo đối tượng Employee
        Employee emp = new Employee();
        //gọi phương thức thành phần
        emp.CallDisplay();
    }
}
```

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



The screenshot shows a Windows Command Prompt window titled "C:\Windows\system32\cmd.exe". The window has a blue title bar and standard Windows window controls (minimize, maximize, close). The command prompt displays the following text:

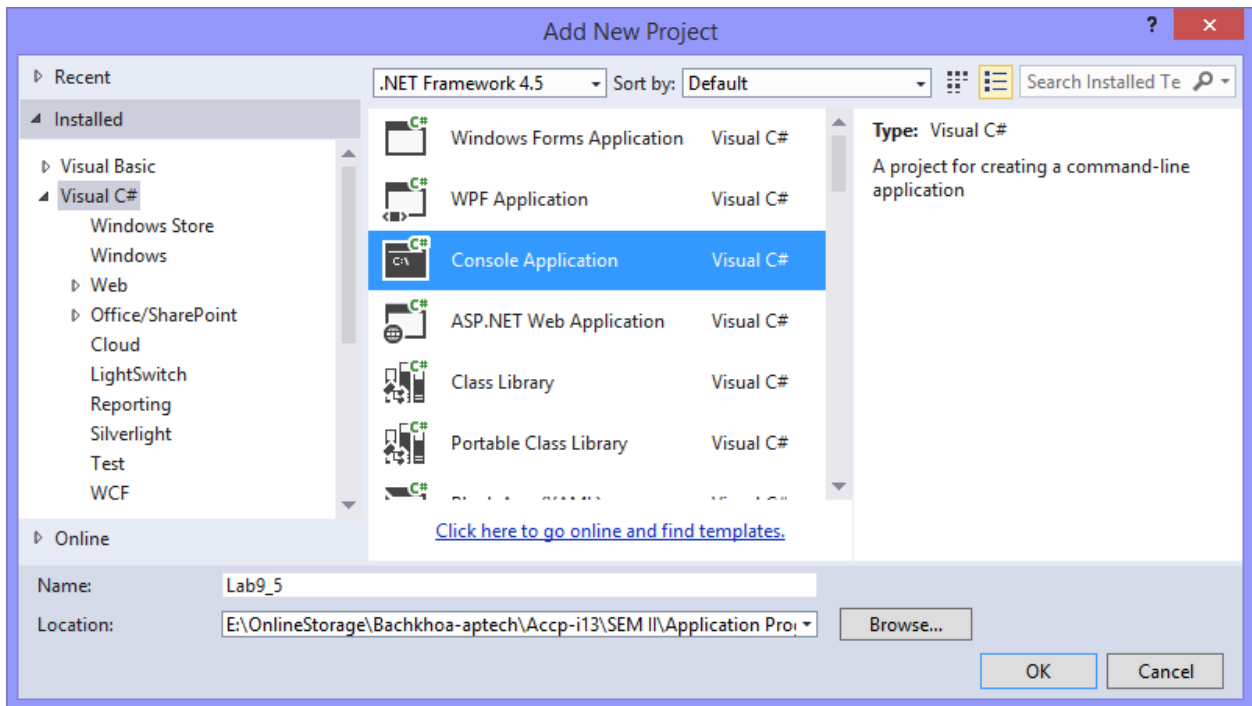
```
Ma so:111  
Ho va ten:Tran Hai Ha  
Dia chi:Ha Noi  
Tuoi:22  
Press any key to continue . . .
```

The text is displayed in a monospaced font. The window has a scrollbar on the right side.

Bài 9.5

Viết chương trình C# minh họa việc khai báo và sử dụng các biến chứa kiểu null.

BƯỚC 1: Kích chuột phải vào Solution “Session9” chọn Add -> New Project ->nhập tên.

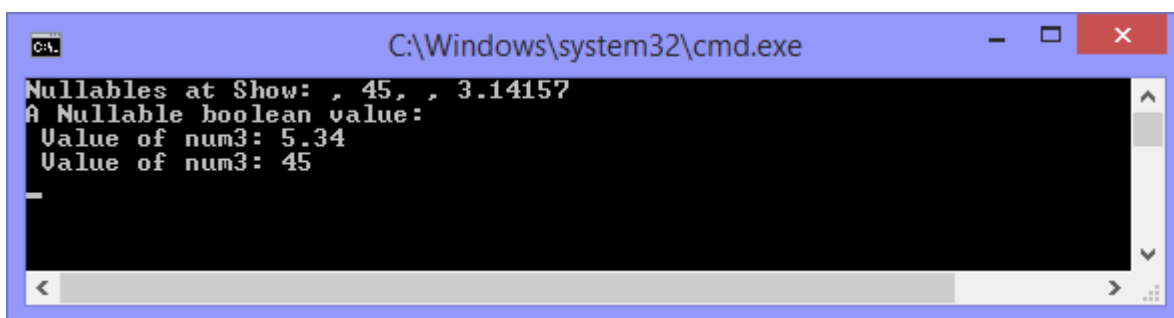


BƯỚC 2: Mở tệp Program.cs và code theo gợi ý sau:

```
static void Main(string[] args)
{
    //khai báo các biến có khả năng chứa dữ liệu null
    int? num1 = null;
    int? num2 = 45;
    double? num3 = new double?();
    double? num4 = 3.14157;
    bool? boolval = new bool?();
    // Hiển thị giá trị
    Console.WriteLine("Nullables at Show: {0}, {1}, {2}, {3}", num1, num2,
num3, num4);
    Console.WriteLine("A Nullable boolean value: {0}", boolval);
}
```

```
//sử dụng toán tử kiểm tra xem biến có null không?  
num3 = num1 ?? 5.34;  
Console.WriteLine(" Value of num3: {0}", num3);  
num3 = num2 ?? 5.34;  
Console.WriteLine(" Value of num3: {0}", num3);  
Console.ReadLine();  
}
```

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



The screenshot shows a Windows Command Prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is as follows:

```
Nullables at Show: , 45, , 3.14157  
A Nullable boolean value:  
Value of num3: 5.34  
Value of num3: 45
```

Phần II Bài tập tự làm

Bài 9.1

Viết chương trình C# minh họa việc tạo và sử dụng phương thức nặc danh như sau:

- Tạo phương thức nặc danh để kiểm tra một số có phải là số nguyên tố hay không?
- Tạo phương thức nặc danh tính tổng các số từ $n \rightarrow m$.

Bài 9.2

Viết chương trình C# minh họa việc tạo và sử dụng phương thức mở rộng như sau:

- Viết một phương thức mở rộng thêm 1 phần tử vào lớp `List<T>` và kiểm tra trùng lặp, nếu phần tử đó tồn tại thì không cho thêm vào nữa.

HẾT