

## **Programming in C#**

### **Mục tiêu:**

Sau khi thực hành xong bạn nắm được các khái niệm:

*Namespace*

*Exception Handling*

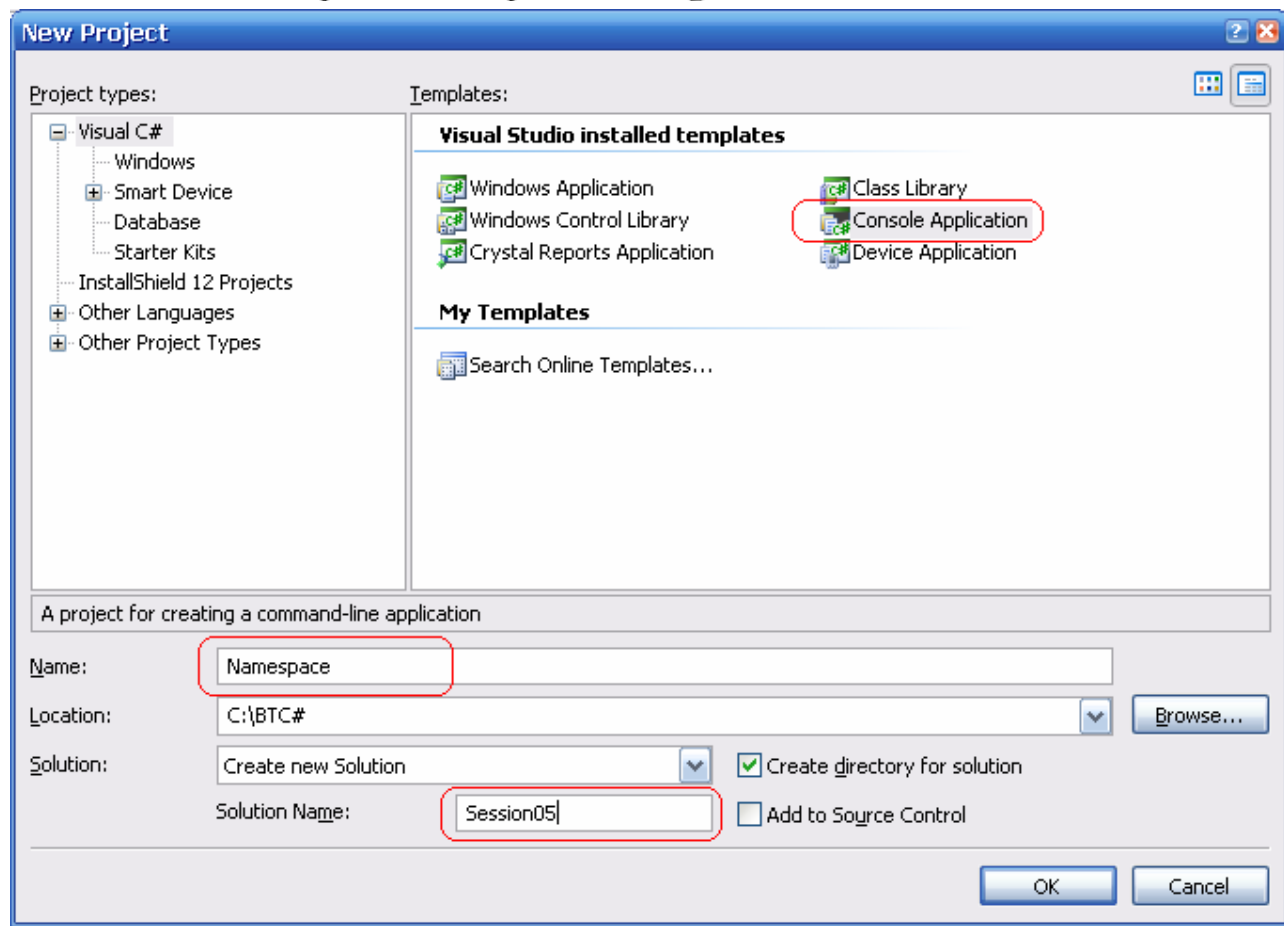
## **Phần I: Thực hành theo từng bước – 45 phút**

### **Bài tập 1.1: Using namespace**

Step 1: Open Visual Studio

Step 2: Select the menu File->New->Project to create console based project named 'Namespace' and

Solution named Session05 as following



Step 3: Rename the class file 'program.cs' to 'Namespace.cs'

Step 4: Replace code in 'Namespace.cs' with given code

```
using System;
using Customer;
using Order;
namespace Customer{
    class Cust_details{
        public string strName;
        public void getName(){
            Console.WriteLine("Enter your name :");
            strName = Console.ReadLine();
        }
    }
}
namespace Order{
    class Grocery_items{
        public void Ord_grocery(){
            Cust_details objCust1 = new Cust_details();
            objCust1.getName();
            Console.WriteLine("Hello {0}", objCust1.strName);
            Console.WriteLine("You have ordered grocery items");
        }
    }
    class Bakery_items{
        public void Ord_bakery(){
            Cust_details objCust2 = new Cust_details();
            objCust2.getName();
            Console.WriteLine("Hello {0}", objCust2.strName);
            Console.WriteLine("You have ordered bakery items");
        }
    }
}
class OrderTest{
    public static void Main(){
        string choice;
        Console.WriteLine("What would you like to order? 1-Grocery Items,
            2- Bakery Items");
        choice = Console.ReadLine();
        if (choice == "1"){
            Grocery_items objGrocery = new Grocery_items();
            objGrocery.Ord_grocery();
        }
        else{
            if (choice == "2"){
                Bakery_items objBakery = new Bakery_items();
                objBakery.Ord_bakery();
            }
            else{
                Console.WriteLine("Enter either 1 or 2");
            }
        }
    }
}
```

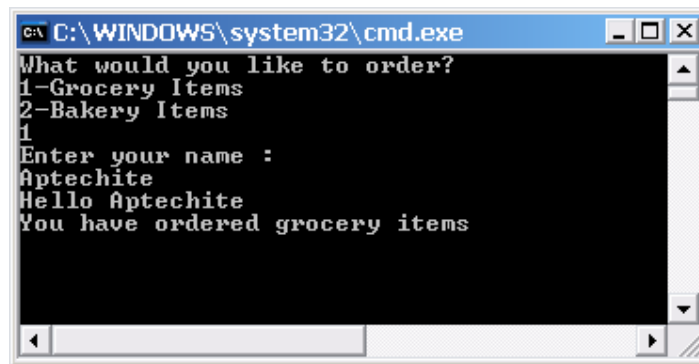
```
        }  
    }  
    Console.ReadLine();  
}  
}
```

Step 5: Select menu File -> Save to save the file

Step 6: Select Build -> Build 'Namespace.cs' option to build the project

Step 7: Select Debug -> Start without Debugging to execute the program

The output of the program as following



```
C:\WINDOWS\system32\cmd.exe  
What would you like to order?  
1-Grocery Items  
2-Bakery Items  
1  
Enter your name :  
Aptechite  
Hello Aptechite  
You have ordered grocery items
```

## Bài tập 1.2: Exception Handling

Step 1: Add a console based project 'ExHandling1' to the solution Step

2: Right click on project 'ExHandling1' -> set as Startup project Step

3: Rename the class file 'Program.cs' to 'ExHandling1.cs'

Step 4: Replace the code in 'ExHandling1.cs' with the given code

```
using System;  
using System.Collections.Generic;  
using System.Text;  
  
namespace Bai05{  
    class Vidul{  
        static void Main(string[] args){  
            byte[] a = new byte[5];  
  
            //nhap mang  
            try{  
                for (int i = 0; i <= 5; i++){
```

```
        Console.WriteLine("a[{0}]=", i + 1);
        a[i] = Convert.ToByte(Console.ReadLine());
    }
}
catch (FormatException ex){
    //Console.WriteLine(ex.Message); Console.WriteLine("Khong duoc
    nhap ki tu cho mang so");
}
catch (OverflowException ex){
    //Console.WriteLine(ex.Message);
    Console.WriteLine("Khong duoc nhap gia tri nam ngoai mien
    0-255");
}
catch (IndexOutOfRangeException ex){
    //Console.WriteLine(ex.Message); Console.WriteLine("Loi vuot qua
    pham vi cua mang");
}

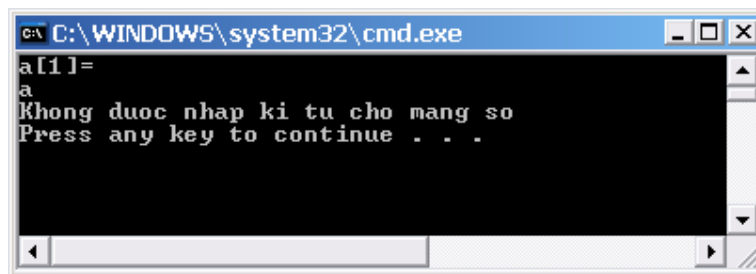
//in mang
for (int i = 0; i < 5; i++)
    Console.Write(" {0}", a[i]);
}
}
```

Step 5: Select menu File -> Save to save the file

Step 6: Select Build -> Build 'ExHandling1' option to build the project

Step 7: Select Debug -> Start without Debugging to execute the program

The output of program as following



**Bài tập 1.3: Write a program to accept a number and print multiplication table of that number. Use exception handling to ensure that user enters only numeric values and the number entered is greater than zero.**

## *C# - Lecture 2 – Namespace and Exception Handling*

Step 1: Add a console based project 'ExHandling2' to the solution Step

2: Right click on project 'ExHandling2' -> set as Startup project Step

3: Rename the class file 'Program.cs' to 'ExHandling1.cs'

Step 4: Replace the code in 'ExHandling2.cs' with the given code

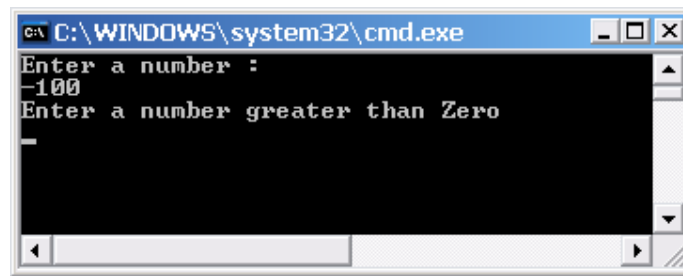
```
using System;
public class InvalidInput : ApplicationException{
    public InvalidInput()
        : base("Enter a number greater than Zero") {}
}
class TestExcep{
    public static void Main(){
        int intCnt;
        int intNum = 0; Console.WriteLine("Enter a number :");
        try{
            intNum = Convert.ToInt32(Console.ReadLine());
            if (intNum <= 0){
                throw new InvalidInput();
            }
        }
        catch (InvalidInput objInvalidInput){
            Console.WriteLine(objInvalidInput.Message);
        }
        catch (System.FormatException objFormatException){
            Console.WriteLine(objFormatException.Message);
        }
        finally{
            if (intNum > 0){
                for (intCnt = 1; intCnt <= 10; intCnt++){
                    Console.WriteLine(intCnt * intNum);
                }
            }
            Console.ReadLine();
        }
    }
}
```

Step 5: Select menu File -> Save to save the file

Step 6: Select Build -> Build 'ExHandling2' option to build the project

Step 7: Select Debug -> Start without Debugging to execute the program

The output of program as following



```
C:\WINDOWS\system32\cmd.exe
Enter a number :
-100
Enter a number greater than Zero
_
```

## Bài tập 1.4: Throw statement

Step 1: Add a console based project 'ExHandling3' to the solution Step

2: Right click on project 'ExHandling3' -> set as Startup project Step 3:

Rename the class file 'Program.cs' to 'ExHandling1.cs'

Step 4: Replace the code in 'ExHandling3.cs' with the given code

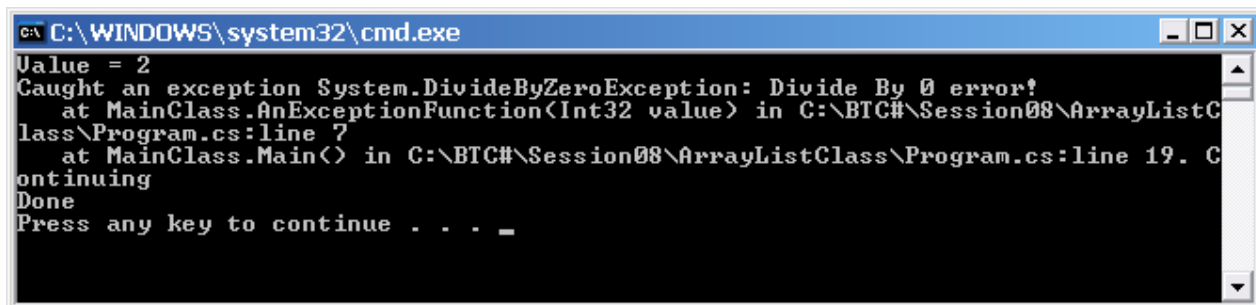
```
using System;
class MainClass{
    public static int AnExceptionFunction(int value){
        if (value == 0) // Can't divide by zero
            throw new DivideByZeroException("Divide By 0 error!");
        int x = 20 / value;
        return x;
    }
    public static void Main() {
        int value = 0;
        try{
            value = AnExceptionFunction(10); // This works ok
            Console.WriteLine("Value = {0}", value);
            AnExceptionFunction(0); // This doesn't
            Console.WriteLine("Value = {0}", value);
        }
        catch (Exception e){
            Console.WriteLine("Caught an exception {0}. Continuing", e);
        }
        Console.WriteLine("Done");
    }
}
```

Step 5: Select menu File -> Save to save the file

Step 6: Select Build -> Build 'ExHandling3' option to build the project

Step 7: Select Debug -> Start without Debugging to execute the program

The output of program as following



```
C:\WINDOWS\system32\cmd.exe
Value = 2
Caught an exception System.DivideByZeroException: Divide By 0 error!
   at MainClass.AnExceptionFunction(Int32 value) in C:\BTC#\Session08\ArrayListClass\Program.cs:line 7
   at MainClass.Main() in C:\BTC#\Session08\ArrayListClass\Program.cs:line 19. Continuing
Done
Press any key to continue . . . _
```



## Bài tập 1.5: Finally statement

Step 1: Add a console based project 'FinallyStmt' to the solution

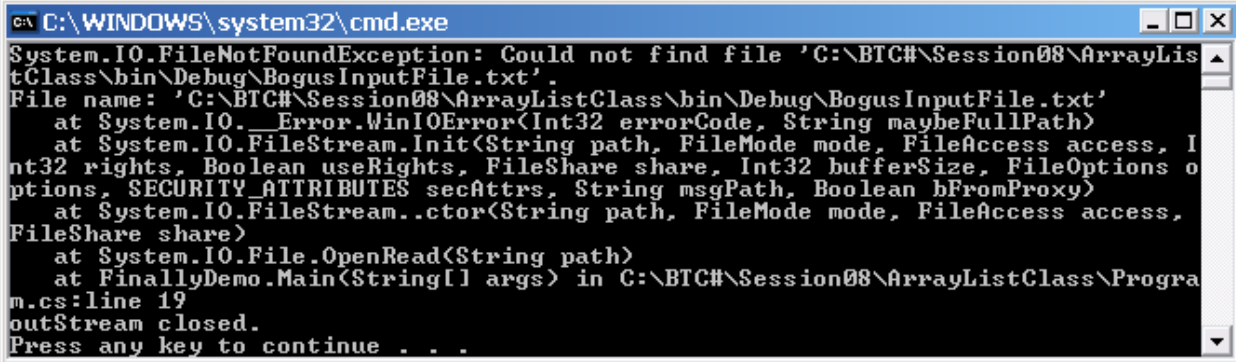
Step 2: Right click on project 'FinallyStmt' -> set as Startup project

Step 3: Rename the class file 'Program.cs' to 'FinallyStmt.cs'

Step 4: Replace the code in 'FinallyStmt.cs' with the given code

```
using System;
using System.IO;
class FinallyDemo{
    static void Main(string[] args){
        FileStream outputStream = null;
        FileStream inputStream = null;
        try{
            //mo file de ghi du lieu
            outputStream = File.OpenWrite("DestinationFile.txt");
            //mo file de doc du lieu
            inputStream = File.OpenRead("BogusInputFile.txt");
            //cac cau lenh doc du lieu tu file
        }
        catch (Exception ex){
            Console.WriteLine(ex.ToString());
        }
        finally{
            if (outputStream != null){
                outputStream.Close();
                Console.WriteLine("outStream closed.");
            }
            if (inputStream != null){
                inputStream.Close();
                Console.WriteLine("inStream closed.");
            }
        }
    }
}
```

The output of program as following



```
C:\WINDOWS\system32\cmd.exe
System.IO.FileNotFoundException: Could not find file 'C:\BTC#\Session08\ArrayListClass\bin\Debug\BogusInputFile.txt'.
File name: 'C:\BTC#\Session08\ArrayListClass\bin\Debug\BogusInputFile.txt'
   at System.IO.__Error.WinIOError(Int32 errorCode, String maybeFullPath)
   at System.IO.FileStream.Init(String path, FileMode mode, FileAccess access, Int32 rights, Boolean useRights, FileShare share, Int32 bufferSize, FileOptions options, SECURITY_ATTRIBUTES secAttrs, String msgPath, Boolean bFromProxy)
   at System.IO.FileStream..ctor(String path, FileMode mode, FileAccess access, FileShare share)
   at System.IO.File.OpenRead(String path)
   at FinallyDemo.Main(String[] args) in C:\BTC#\Session08\ArrayListClass\Program.cs:line 19
outStream closed.
Press any key to continue . . .
```

## Phần II: Tự thực hành – 60 phút

### Bài tập 2.1: Tạo namespace

Tạo một namespace gọi là **CustomerSpace** và trong namespace có 1 lớp **Customer** với phương thức cập nhật **customer name**. Tạo một namespace khác gọi là **OrderSpace** và tạo hai lớp trong đó. Lớp **GroceryOrder** cho các mặt hàng tạp hóa và lớp **BreadOrder** cho các sản phẩm bánh mì. Chương trình **Main()** có thể cập nhật **customer name** và có một chọn lựa khi nào một **customer** đặt hàng tạp hóa hoặc sản phẩm bánh mì. Theo đó, lớp phù hợp sẽ được gọi và một thông điệp hiển thị thông báo với người sử dụng về lựa chọn của mình.

### Bài tập 2.2: Xử lý Exception

Viết một Exception có tên là **AmountException** để kiểm tra các điều kiện sau:

Khi một trợ giảng Senior nhận lương ít hơn 60,000

Khi hoa hồng (bonus) lớn hơn 10,000

Lớp **Exception** có một trường tên là **personName**.

Viết một chương trình **Test** để làm việc với các đối tượng khác nhau và hành vi của lớp đó.

Thêm một số mã nguồn để mô phỏng tính đa hình. Đồng thời hiển thị chức năng của lớp Exception của bạn bằng cách thêm một số mã nguồn thích hợp