

## LAB 1

### MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Sử dụng Static class, Partial class
- ✓ Tạo được các ứng dụng sử dụng Generic Namespace

### Bài 1 (2 điểm)

Sử dụng static class khai báo lớp “MyCollege” gồm 2 static field là “CollegeName” và “Address”. Khởi tạo thông tin cho các field thông qua static constructor và xuất các thông tin lên màn hình console.

Gợi ý:

```
static class MyCollege
{
    //static fields
    public static string CollegeName;
    public static string Address;

    //static constructor
    0 references
    static MyCollege()
    {
        CollegeName = "FPT Polytechnic";
        Address = "391A Nam Kỳ Khởi Nghĩa, Q. 3, TP. Hồ Chí Minh";
    }
}
0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        Console.WriteLine(MyCollege.CollegeName);
        Console.WriteLine(MyCollege.Address);
        Console.Read();
    }
}
```

**Bài 2 (2 điểm)**

Xây dựng class Student gồm các thông tin StudentName, Course, CollegeName, CollegeAddress. Bổ sung các phần còn thiếu vào đoạn code bên dưới để hoàn thiện ứng dụng:

```
class Student
{
    //non-static data members
    StudentName;
    Course;
    2 references
    public void SetStudentDetails(string StuName, string Cou)
    {
        .... = StuName;
        .... = Cou;
    }
    2 references
    public void DisplayStudentDetails()
    {
        Console.WriteLine(StudentName + " - " + Course);
    }

    //static data members
    public CollegeName = "FPT PolyTechnic";
    public CollegeAddress = "391A Nam Kỳ Khởi Nghĩa, Phường 8, Quận 3, Hồ Chí Minh";

    //static methods
    0 references
    public static void DisplayCollegeDetails()
    {
        Console.WriteLine(CollegeName);
        Console.WriteLine(CollegeAddress);
    }
}
```

```

class Program
{
    0 references
    static void Main(string[] args)
    {
        //access static members
        DisplayCollegeDetails();

        //access non-static members
        Console.WriteLine();
        Student s1 = new Student();
        Student s2 = new Student();
        s1.SetStudentDetails("Thepv", "Phd");
        s2.SetStudentDetails("Nghiemn", "MBA");
        s1.DisplayStudentDetails();
        s2.DisplayStudentDetails();
        Console.Read();
    }
}
  
```

### Bài 3 (2 điểm)

Viết chương trình sử dụng partial class cho các phép tính cộng trừ nhân chia.

Gợi ý:

- Xây dựng Partial Class 1

```

partial class clsCalculation
{
    1 reference
    public void Addition(int a, int b)
    {
        Console.WriteLine("Output Addition is {0}", a + b);
    }

    1 reference
    public void Substraction(int a, int b)
    {
        Console.WriteLine("Output Substraction is {0}", a - b);
    }
}
  
```

- Xây dựng Partial Class 2

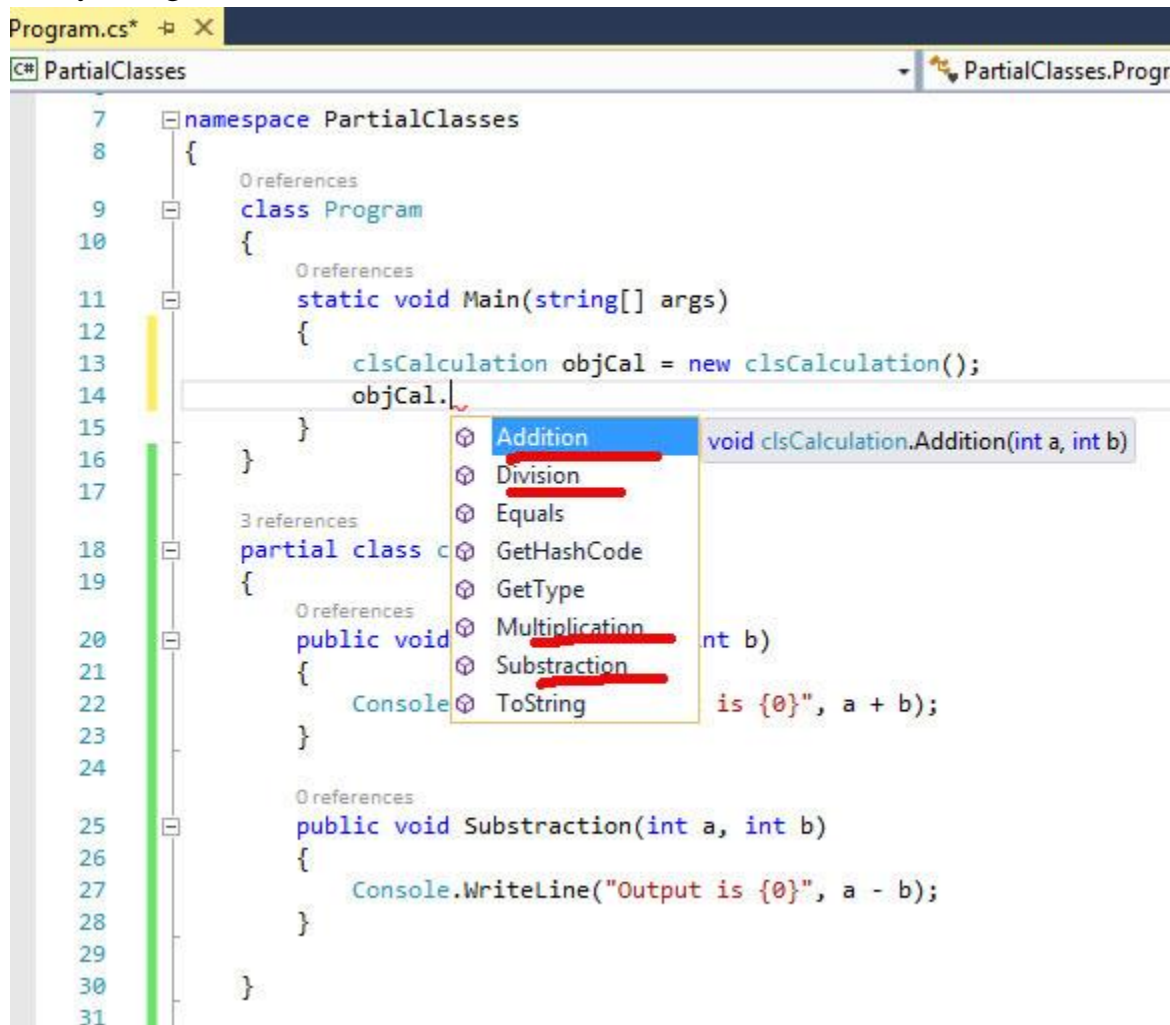
```

partial class clsCalculation
{
    1 reference
    public void Multiplication(int a, int b)
    {
        Console.WriteLine("Output Multiplication is {0}", a * b);
    }

    1 reference
    public void Division(int a, int b)
    {
        Console.WriteLine("Output Division is {0}", a / b);
    }
}

```

- Xử lý trong Main



**Bài 4 (2 điểm)**

Viết chương trình so sánh hiệu suất giữa List và HashSet

```

static void Main(string[] args)
{
    int times = 10000000;
    for (int listSize = 1; listSize < 10; listSize++){
        List<string> list = new List<string>();
        HashSet<string> hashset = new HashSet<string>();
        for (int i = 0; i < listSize; i++)
        {
            list.Add("string" + i.ToString());
            hashset.Add("string" + i.ToString());
        }
        Stopwatch timer = new Stopwatch();
        timer.Start();
        for (int i = 0; i < times; i++){
            list.Remove("string0");
            list.Add("string0");
        }
        timer.Stop();
        Console.WriteLine(listSize.ToString() + " item LIST strs time: " + timer.ElapsedMilliseconds.ToString() + "ms");
        timer = new Stopwatch();
        timer.Start();
        for (int i = 0; i < times; i++){
            hashset.Remove("string0");
            hashset.Add("string0");
        }
        timer.Stop();
        Console.WriteLine(listSize.ToString() + " item HASHSET strs time: " + timer.ElapsedMilliseconds.ToString() + "ms");
        Console.WriteLine();
    }
}

for (int listSize = 1; listSize < 10; listSize += 3){
    List<object> list = new List<object>();
    HashSet<object> hashset = new HashSet<object>();
    for (int i = 0; i < listSize; i++)
    {
        list.Add(new object());
        hashset.Add(new object());
    }
    object objToAddRem = list[0];
    Stopwatch timer = new Stopwatch();
    timer.Start();
    for (int i = 0; i < times; i++)
    {
        list.Remove(objToAddRem);
        list.Add(objToAddRem);
    }
    timer.Stop();
    Console.WriteLine(listSize.ToString() + " item LIST objs time: " + timer.ElapsedMilliseconds.ToString() + "ms");
    timer = new Stopwatch();
    timer.Start();
    for (int i = 0; i < times; i++){
        hashset.Remove(objToAddRem);
        hashset.Add(objToAddRem);
    }
    timer.Stop();
    Console.WriteLine(listSize.ToString() + " item HASHSET objs time: " + timer.ElapsedMilliseconds.ToString() + "ms");
    Console.WriteLine();
}
Console.ReadLine();
    
```

**Bài 5 (2 điểm)**

Giảng viên cho thêm