

LAB 2

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Sử dụng Implicitly, Dynamic, Nullable, Anonymous typed

Bài 1 (4 điểm)

Dùng từ khóa var hoặc dynamic và fix bug hoàn thiện các đoạn code sau :

a/

```
public class ImplicitlyVariable
{
    var imp = 15;
    // Main method
    0 references
    static public void Main()
    {
        // trying to print the value of 'imp'
        Console.WriteLine(imp);
    }
}
```

b/

```
class Program
{
    0 references
    static void Main()
    {
        ...data = new...<string, int>();
        data.Add("cat", 2);
        data.Add("dog", 1);
        Console.WriteLine("cat - dog = {0}",
            data["cat"] - data["dog"]);
    }
}
```

c/

```

class Program
{
    0 references
    static void Main(string[] args)
    {
        GetDetails(100);
        GetDetails("Welcome to FPoly");
        GetDetails(true);
        GetDetails(20.50);
        Console.ReadLine();
    }
    4 references
    static void GetDetails(d)
    {
        Console.WriteLine(d);
    }
}
    
```

d/

```

class Program
{
    0 references
    static void Main(string[] args)
    {
        dobj = new User();
        dobj.GetInfo();
    }
}
1 reference
class User
{
    0 references
    public int Id { get; set; }
    0 references
    public string Name { get; set; }
    0 references
    public void GetDetails(d)
    {
        Console.WriteLine(d);
    }
}
    
```

Bài 2 (4 điểm)

Sử dụng Anonymous typed, Anonymous method và fix bug hoàn thiện các đoạn code sau:

a/

```
class Program
{
    0 references
    static void Main(string[] args)
    {
        // Create anonymous type object
        userInfo = new
        {
            Id = 1,
            Name = "Suresh Dasari",
            IsActive = true
        };
        // Access anonymous type object properties
        Console.WriteLine("Id:" + Id);
        Console.WriteLine("Name:" + Name);
        Console.WriteLine("IsActive:" + IsActive);
        Console.ReadLine();
    }
}
```

b/

```
class Program
{
    0 references
    static void Main(string[] args)
    {
        // Create Nested anonymous type object
        user = new
        {
            Id = 1,
            Name = "Suresh Dasari",
            IsActive = true,
            jobInfo = { Designation = "Lead", Location = "Hyderabad" }
        };
        // Access anonymous type object properties
        Console.WriteLine("Id: " + user.Id);
        Console.WriteLine("Name: " + user.Name);
        Console.WriteLine("IsActive: " + user.IsActive);
        Console.WriteLine("Designation: {0}, Location: {1}", user.jobInfo.Designation, user.jobInfo.Location);
        Console.ReadLine();
    }
}
```

c/

```

class Program
{
    // Create a delegate
    1 reference
    public void MathOps(int a);
    0 references
    static void Main(string[] args)
    {
        int y = 10;
        // Instantiate the delegate using an anonymous method
        MathOps ops = (int x)
        {
            Console.WriteLine("Add Result: {0}", x + y);
            Console.WriteLine("Subtract Result: {0}", x - y);
        };
        ops(90);
        Console.ReadLine();
    }
}
    
```

Bài 3 (2 điểm)

Giảng viên cho thêm