

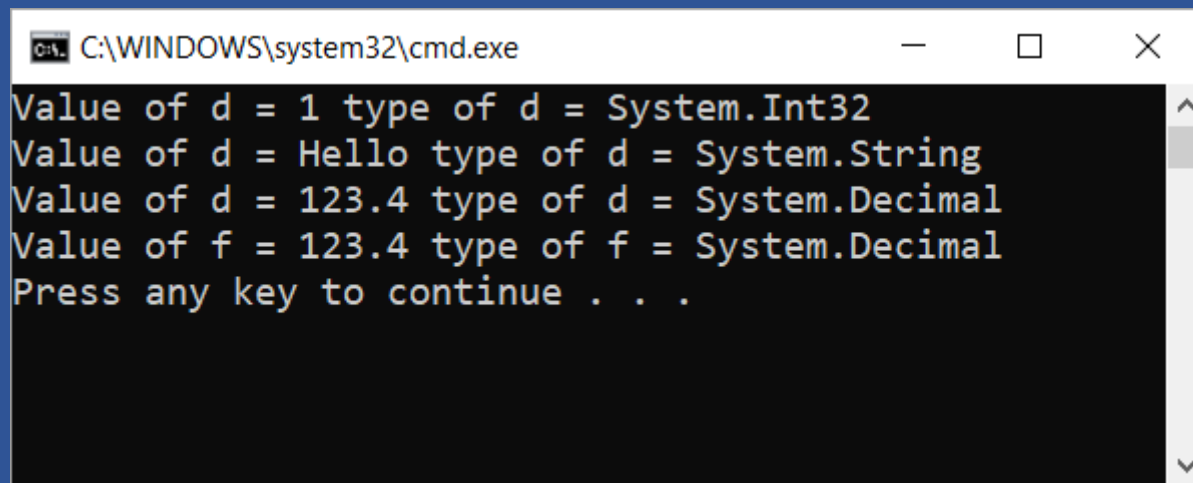
Dynamic

Dynamic type

- Introduced in C# 4
- Is a static type that bypasses type checking
- Function like type “Object”
- Support any operation form COM API, IronPython

Exercise 00010 : Simple example

```
dynamic d = 1;  
Console.WriteLine($"Value of d = {d} type of d = {d.GetType()}");  
d = "Hello";  
Console.WriteLine($"Value of d = {d} type of d = {d.GetType()}");  
d = 123.4m;  
Console.WriteLine($"Value of d = {d} type of d = {d.GetType()}");  
int i = 0;  
var f = d + i;  
Console.WriteLine($"Value of f = {d} type of f = {d.GetType()}");
```

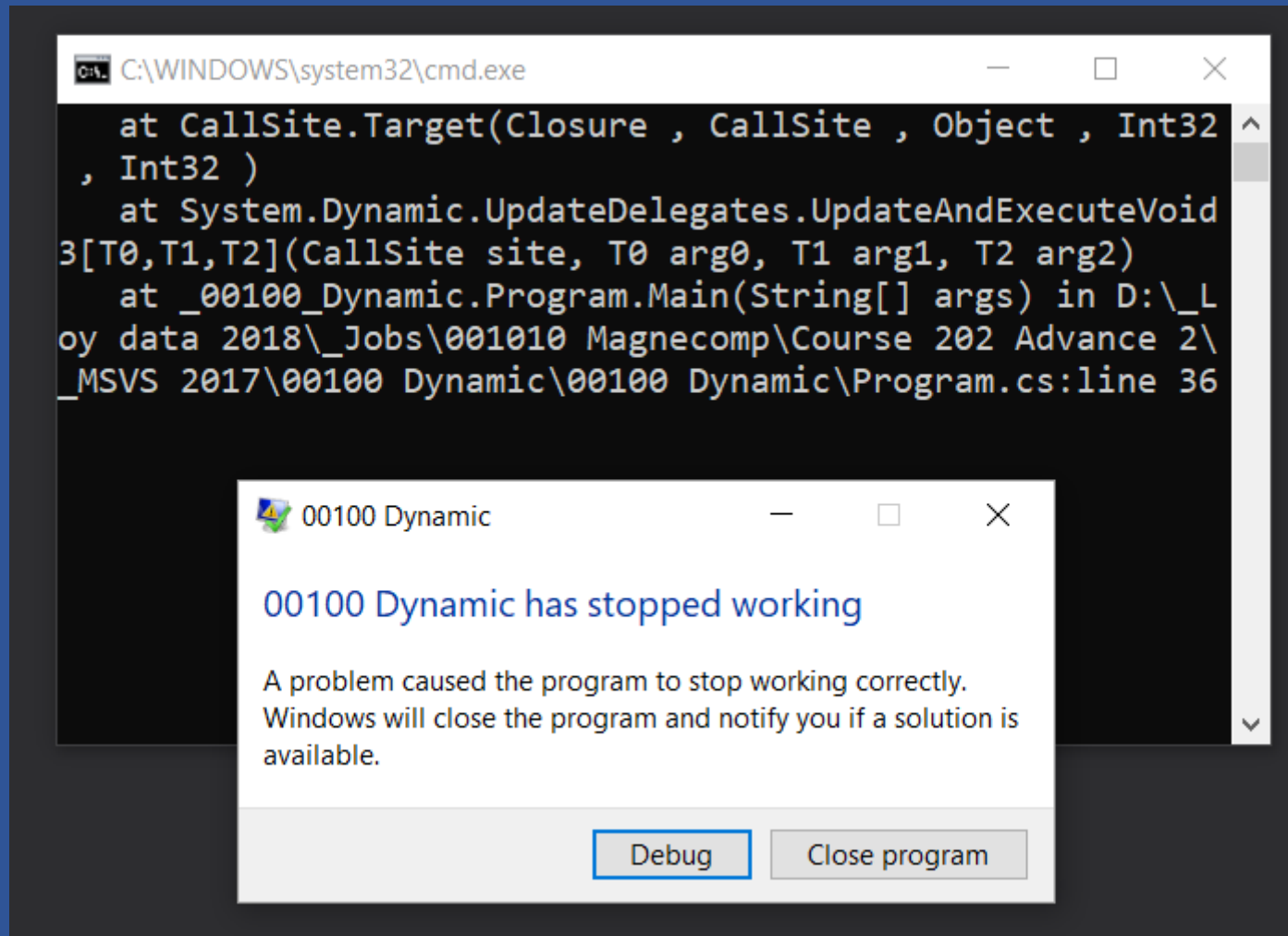


The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The output of the C# program is displayed as follows:

```
Value of d = 1 type of d = System.Int32  
Value of d = Hello type of d = System.String  
Value of d = 123.4 type of d = System.Decimal  
Value of f = 123.4 type of f = System.Decimal  
Press any key to continue . . .
```

Exercise 00100 : Dynamic support any operation.

Line 36 causes no compile error (exampleMethod1 has only one param)



Exercise 00110 : Dynamic and COM interop

- need Nuget Microsoft.Office.Interop.Excel
- and excel file person001.xlsx

```
// Get fully qualified path for xlsx file
var spreadsheetLocation = @"d:\temp\person001.xlsx";
var exApp = new Excel.Application();
var exWbk = exApp.Workbooks.Open(spreadsheetLocation);
var exWks = (Excel.Worksheet)exWbk.Sheets["Sheet1"];
Excel.Range xlRange = exWks.UsedRange;
```