

/\*\*\*\*\*\*

## EXPERIMENT NO: 05 (GROUP – B)

**TITLE** : Write a program to create Dynamic Link Library for any mathematical operation and write an application program to test it. (Java Native Interface / Use VB or VC++).

\*\*\*\*\* /

### 1. DLL:-

A dynamic link library (DLL) is a collection of small programs that can be loaded when needed by larger programs and used at the same time. The small program lets the larger program communicate with a specific device, such as a printer or scanner. It is often packaged as a DLL program, which is usually referred to as a DLL file. DLL files that support specific device operation are known as device drivers.

#### Advantages of DLL:-

- Saves memory and reduces swapping. Many processes can use a single DLL simultaneously, sharing a single copy of the DLL in memory. In contrast, Windows must load a copy of the library code into memory for each application that is built with a static link library.
- Saves disk space. Many applications can share a single copy of the DLL on disk. In contrast, each application built with a static link library has the library code linked into its executable image as a separate copy.
- Upgrades to the DLL are easier. When the functions in a DLL change, the applications that use them do not need to be recompiled or relinked as long as the function arguments and return values do not change. In contrast, statically linked object code requires that the application be relinked when the functions change.
- Provides after-market support. For example, a display driver DLL can be modified to support a display that was not available when the application was shipped.
- Supports multi language programs. Programs written in different programming languages can call the same DLL function as long as the programs follow the function's calling convention. The programs and the DLL function must be compatible in the following ways: the order in which the function expects its arguments to be pushed onto the stack, whether the function or the application is responsible for cleaning up the stack, and whether any arguments are passed in registers.
- Provides a mechanism to extend the MFC library classes. You can derive classes from the existing MFC classes and place them in an MFC extension DLL for use by MFC applications.
- Eases the creation of international versions. By placing resources in a DLL, it is much easier to create international versions of an application. You can place the strings for each language version of your application in a separate resource DLL and have the different language versions load the appropriate resources.

- ✓ A potential disadvantage to using DLLs is that the application is not self-contained; it depends on the existence of a separate DLL module.

## 1.2 VB PROGRAM

Public Class Form1

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
    TextBox3.Text = adddll.mul(Val(TextBox1.Text), Val(TextBox2.Text))
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
    TextBox3.Text = adddll.substraction(Val(TextBox1.Text), Val(TextBox2.Text))
End Sub
```

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
    TextBox3.Text = adddll.add(Val(TextBox1.Text), Val(TextBox2.Text))
End Sub
```

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
    TextBox3.Text = adddll.div(Val(TextBox1.Text), Val(TextBox2.Text))
End Sub
End Class
```

## 1.3 DLL PROGRAM

Public Module addm

```
Public Function add(ByVal n1 As Integer, ByVal n2 As Integer)
    Return (n1 + n2)
End Function
Public Function substraction(ByVal n1 As Integer, ByVal n2 As Integer)
    Return (n1 - n2)
End Function
Public Function mul(ByVal n1 As Integer, ByVal n2 As Integer)
    Return (n1 * n2)
End Function
Public Function div(ByVal n1 As Integer, ByVal n2 As Integer)
    Return (n1 / n2)
End Function
End Module
```

**////OUTPUT :**

The image shows a screenshot of a Windows application window titled "Form1". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main content area is light gray and contains a simple calculator interface. It features two input fields labeled "num1" and "num2" with values "12" and "65" respectively. Below these are four buttons labeled "add", "sub", "mul", and "div". The "add" button is highlighted with a blue border. At the bottom, there is a label "result" and an input field containing the value "77".

Label	Value
num1	12
num2	65
add	add
sub	sub
mul	mul
div	div
result	77