

40 pts

Name: \_\_\_\_\_

Class Day / Time: \_\_\_\_\_

Due Date: \_\_\_\_\_

## Lab #4 – Assembly Program – Add Numbers

This lab is a tutorial for learning how to handle the **Assemble-Link Execute Cycle**. The lab uses the tools require to create and edit your assembly source code, then assemble your code and link it to create an executable version of your code. We will be using MASM (Microsoft Assembler) and TextPad. These tools are already installed in the CS Lab computers.

### To create and save the source code:

Create a folder called Lab1 in your flash drive (for example assume - H: drive), or wherever you will store your assembly language programs. You will create a new folder for each assignment. Open TextPad and type the following program (add number modified from lecture example). **Important:** When you are ready to save the program, **TYPE IN THE FULL PATH USING H: In the File Name BOX as: H:\Lab1\Lab1.asm**

(TextPad does not like domain names, and will not assemble your program if it is associated with a domain name.)

```
; Program Description: Lab#1 – Add Numbers, .....
; Author:
; Creation Date:
```

```
.386
.MODEL FLAT
```

```
ExitProcess PROTO NEAR32 stdcall, dwExitCode:dword
```

```
include io.h
```

```
cr EQU 0dh                ; cr = carriage return
lf EQU 0ah                ; lf = line feed
```

```
.STACK 4096
```

```
.DATA
```

```
szPrompt1    BYTE    "Enter first number: ",0
szPrompt2    BYTE    "Enter second number: ",0
szLabel1     BYTE    "The sum is: ",0
dwNumber1    DWORD    ?                ; numbers to be added
dwNumber2    DWORD    ?
szString     BYTE    16 DUP (?)        ; input string for numbers
szSum        BYTE    12 DUP (0)        ; sum in string form
szNewline    BYTE    cr, lf, 0
```

```

.CODE
_start:

    output    szPrompt1    ; prompt for first number
    input     szString,16  ; input first number as ASCII
    atod      szString     ; convert to integer
    mov       dwNumber1, eax ; and store in memory
    output    szPrompt2    ; repeat for second number
    input     szString,16
    atod      szString
    mov       dwNumber2, eax
    mov       eax, dwNumber1
    add       eax, dwNumber2 ; add second number to first number
    dtoa      szSum, eax    ; convert to ASCII
    output    szLabel1     ; output label and results
    output    szSum
    output    szNewline

    INVOKE ExitProcess, 0

PUBLIC _start
END

```

### To create and save the makefile:

The makefile works with MASM's nmake.exe to assemble and link your program. Create the following file, called **Lab1.mak**:

**ALL: LAB1.EXE**

**CLEAN:**

```

-@erase LAB1.EXE
-@erase LAB1.ILK
-@erase LAB1.PDB
-@erase LAB1.OBJ
-@erase LAB1.LST

```

**LAB1.ASM:**

```

LAB1.OBJ: LAB1.ASM
ml /c /coff /Zi LAB1.ASM

```

**LAB1.EXE: LAB1.OBJ**

```

link /debug /subsystem:console /out:LAB1.EXE /entry:start LAB1.OBJ KERNEL32.LIB IO.OBJ

```

Save the file in the Lab1 folder. (Once the **H:** drive designation has been associated with the folder, you should be able to save the file directly, without specifying the path.)

**To save the workspace:**

From the file menu, choose Workspace and Save. **Lab1.tws** will be displayed in the File name: box. Click Save.

When you open this project again, you will open the Workspace, using the **H: path**. (File, Workspace, Open, **H:\Lab1\Lab1.tws**)

**To assemble and link the program:**

Choose Tools, Make 32-bit Assembly or CTRL-4. If any errors are listed in the Command Results window, correct them. Before you use Make again, choose Tools, Clean 32-bit Assembly or CTRL-7, and then Tools, Make 32-bit Assembly or CTRL-4.

**To run the program:**

Select Tools, Run 32-bit Assembly or CTRL-5.

**To print the output screen:**

Right-click on the blue title bar on the Windows screen, and choose Edit, Mark. Highlight the output and press ENTER. Paste the output onto the end on Lab1.asm. Print the Lab1.asm. Don't resave Lab1.asm.

**Turn in (STAPLED IN THIS ORDER)**

1. The **FIRST PAGE** of this assignment as a coversheet
2. The listing of Lab1.asm properly documented
3. The listing of Lab1.mak
4. The output from the program, pasted into Lab1.asm