Department of Computer Science
Faculty of Computing
UNIVERSITI TEKNOLOGI MALAYSIA

SUBJECT NAME:	**COMPUTER ORGANIZATION AND ARCHITECTURE**
SUBJECT CODE:	**SECR 1033**
SEMESTER:	**2 - 2023/24**
LAB TITLE:	**Programming 1: Assembly Language Fundamentals**
INSTRUCTION:	**Student is required to have instructor's signature before proceed from each lab work.**

STUDENT INFO :

Name: JOANNE CHING YIN XUAN , CHUA JIA LIN

Metric No: A23CS0227, A23CS0069

Section: 2

Email: joanneyin@graduate.utm.my
chuajialin@graduate.utm.my

SUBMITTED DATE: 25/4/2024

COMMENTS:

Lab # 1

Simple program to familiarize with Program Code, Rebuild & Start Without Debugging

Execute the programs below:
PART 1:

i. **Part A**: Adding and Subtracting Integers

```
TITLE Add and Subtract (AddSub.asm)

; This program adds and subtracts 32-bit integers
; Authors:
; Date:
; Revision:

INCLUDE Irvine32.inc

.data
TOTAL dword 0 ; a variable named TOTAL (declared as DWORD)

.code

main PROC

    mov eax, 123400h ; Set EAX with the value of 123400h
    add eax, 567800h ; Add the content of EAX with 567800h
    sub eax, 77700h ; Subtract content of EAX with 77700h
    mov TOTAL, eax  ; Store content of EAX to TOTAL

    call DumpRegs

    exit
main ENDP

END main
```

Experimental Results:
a) Rebuild & Start Without Debugging

Completed Fully Partially : Checked by: _____

Screenshot Result:

```
  EAX=00613500  EBX=00D17000  ECX=001F10AA  EDX=001F10AA
  ESI=001F10AA  EDI=001F10AA  EBP=00EFFA40  ESP=00EFFA34
  EIP=001F3679  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1


C:\Users\HP\source\repos\COA\lab1\Debug\lab1.exe (process 22036) exited with code 0.
Press any key to close this window . . .
```

ii. **Part B**: Adding Variables

```
TITLE Add and Subtract, Version 2(AddSub2.asm)

; This program adds and subtracts 32-bit unsigned
; integers and stores the sum in a variable.
; Authors:
; Date:
; Revision:

INCLUDE Irvine32.inc
.data
val1 DWORD 10000h
val2 DWORD 40000h
val3 DWORD 20000h
finalVal DWORD ?
.code

main PROC
mov eax,val1     ; start with 10000h
add eax,val2     ; add 40000h
sub eax,val3     ; subtract 20000h
mov finalVal,eax ; store the result (30000h)

call DumpRegs    ; display the registers

exit

main ENDP
END main
```

Experimental Results:

a) Rebuild & Start Without Debugging

Completed Fully Partially : Checked by: _____

Screenshot Result:

```
EAX=00030000  EBX=004E8000  ECX=00BD10AA  EDX=00BD10AA
ESI=00BD10AA  EDI=00BD10AA  EBP=006FFD18  ESP=006FFD0C
EIP=00BD367B  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1


C:\Users\HP\source\repos\COA\lab_template\lab_template\Debug\lab_template.exe (process 13264) exited with code 0.
Press any key to close this window . . .
```

iii. **Part C**: Add and Subtract 8 and 16-Bit Version

```
TITLE Add and Subtract, Version 3
; This program adds and subtracts 8 and 16 bit
; unsigned integers and stores the sum in a variable.
; Authors:
; Date:
; Revision:

INCLUDE Irvine32.inc
.data
valw1 WORD 1000h
valw2 WORD 4000h
valw3 WORD 2000h
finalValw WORD ?

valb1 BYTE 10h
valb2 BYTE 40h
valb3 BYTE 20h
finalValb BYTE ?

.code
main PROC
mov ax,valw1; start with 10000h
add ax,valw2       ; add 40000h
sub ax,valw3       ; subtract 20000h
mov finalValw,ax   ; store the result (30000h)
call DumpRegs      ; display the registers

mov ah,valb1; start with 10000h
add ah,valb2       ; add 40000h
sub ah,valb3       ; subtract 20000h
mov finalValb,ah   ; store the result (30000h)
call DumpRegs      ; display the registers

exit
main ENDP
END main
```

Experimental Results:

a) Rebuild & Start Without Debugging

Completed Fully Partially : Checked by: _____

Screenshot Result:

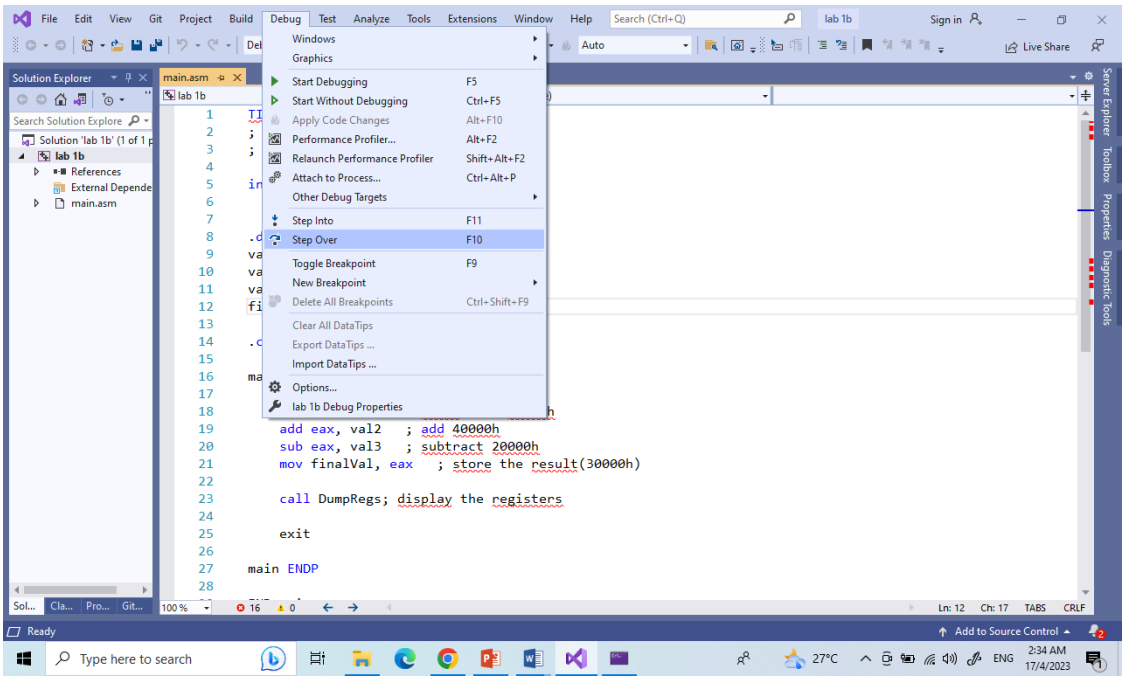
```
EAX=00D73000  EBX=00B63000  ECX=006310AA  EDX=006310AA
ESI=006310AA  EDI=006310AA  EBP=00D7FCF8  ESP=00D7FCEC
EIP=0063367F  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1


EAX=00D73000  EBX=00B63000  ECX=006310AA  EDX=006310AA
ESI=006310AA  EDI=006310AA  EBP=00D7FCF8  ESP=00D7FCEC
EIP=0063369C  EFL=00000206  CF=0  SF=0  ZF=0  OF=0  AF=0  PF=1


C:\Users\Asus\source\repos\lab1c\Debug\lab1c.exe (process 10344) exited with code 0.
Press any key to close this window . . .
```

Detail Debugging Process

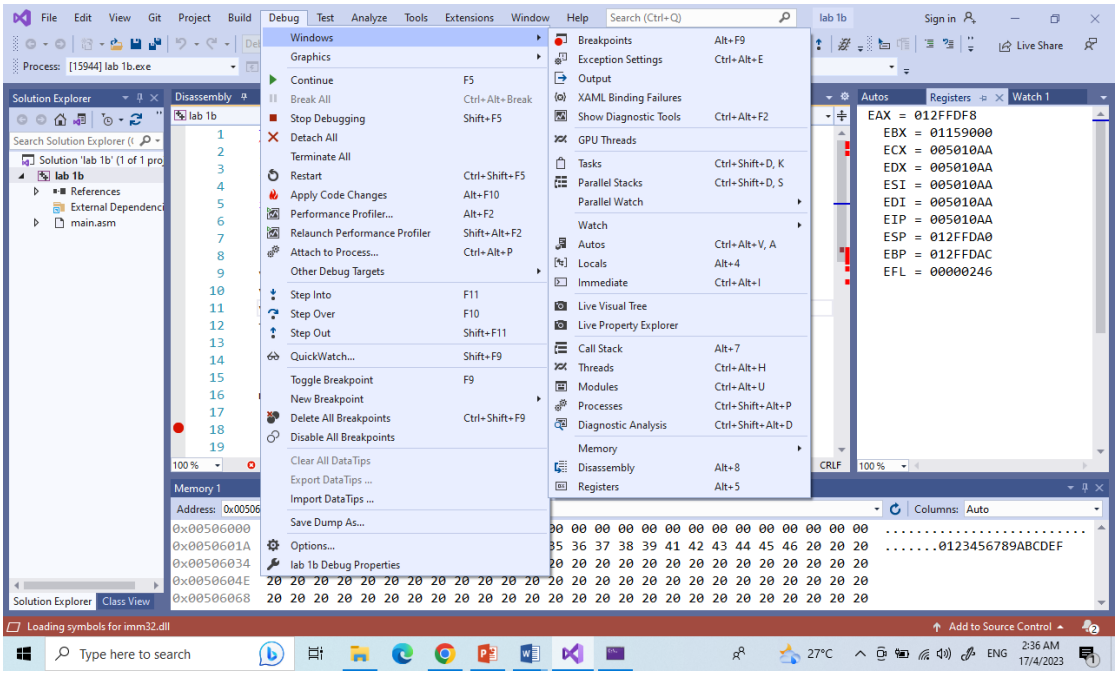
PART 2:

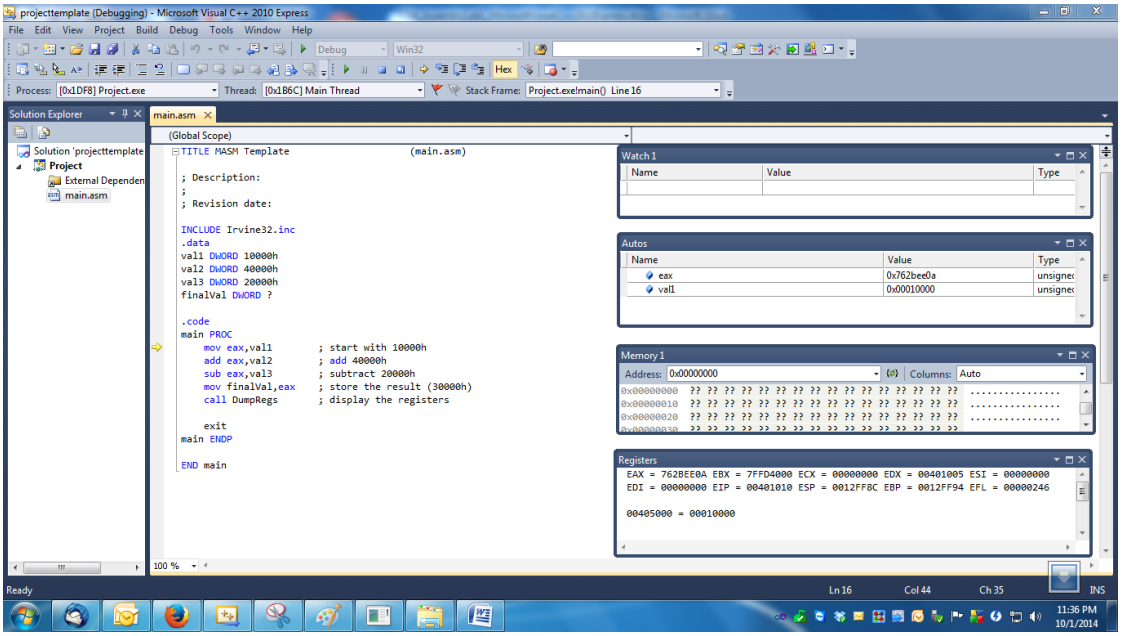
Run Debugging Process for Part B, Part C and Capture Video

1. Press F10 for step by step debugging.

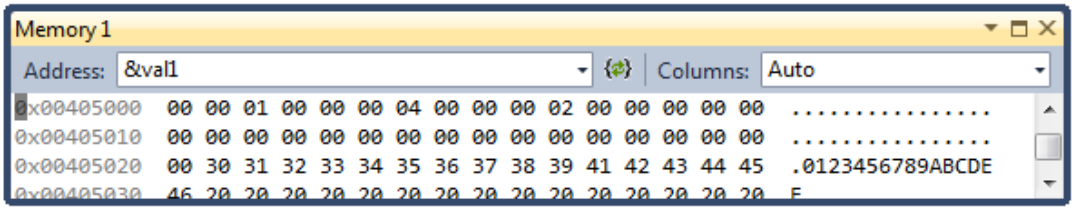


3. Open 5 windows:
   a. Watch
   b. Autos
   c. Memory
   d. Registers
   e. Disassembly

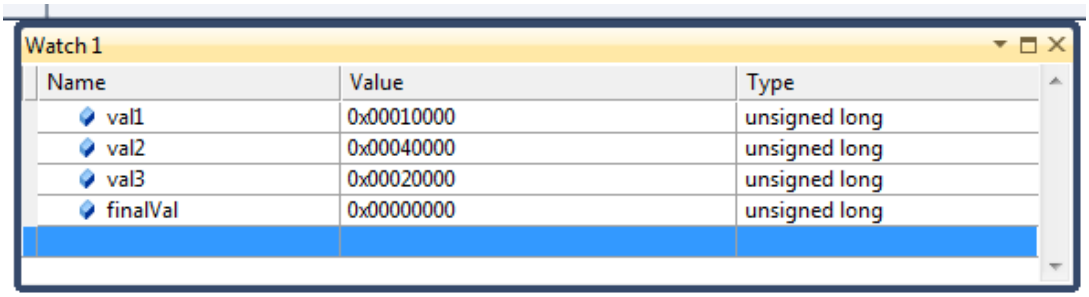




4. Default memory data segment value is at first variable: &val1 or 0x00405000 (depend on memory used).



5. Add variables val1, val2, val3 and finalval to Watch



6. F10 to trace/step the assembly program line by line.

7. Please debug by looking at value changes at Watch, Memory, Autos and Registers windows.

8. Disassembly window can show the following optional information:

- Memory address where each instruction is located. For native applications, it is the actual memory address. For Visual Basic or C#, it's an offset from the beginning of the function.
- Source code from which the assembly code derives.
- Code bytes, that is, the byte representations of the actual machine or MSIL instructions.
- Symbol names for the memory addresses.
- Line numbers corresponding to the source code.

What is Byte Code (or Machine Code) for the following assembly instructions:

```
    mov ax,valw1  ; start with 10000h
    add ax,valw2        ; add 40000h
    sub ax,valw3        ; subtract 20000h
    mov finalValw,ax    ; store the result (30000h)
```

Byte Code:

```
mov ax, valw1; start with 10000h
66 A1 00 60 98 00    mov         ax,word ptr [valw1 (0986000h)]
add ax, valw2; add 40000h
66 03 05 02 60 98 00 add         ax,word ptr [valw2 (0986002h)]
sub ax, valw3; subtract 20000h
66 2B 05 04 60 98 00 sub         ax,word ptr [valw3 (0986004h)]
mov finalValw, ax; store the result(30000h)
66 A3 06 60 98 00    mov         word ptr [finalValw (0986006h)],ax
```

Experimental Results:

a) Rebuild & Start Debugging CamStudio:

Completed Fully Partially : Checked by: _____

Lab 1 Submission

Due: 6 May 2023 (Saturday: 23.59pm)

This is considering as a group submission. Each group requires ONLY 1 submission.

1. Screen capture results part 1(a), (b), (c) - Start Without Debugging

2. Code Submission

i) Zip both your exercise folders;
- [Lab 1a]
- [Lab 1b]
- [Lab 1c]

ii) Combine into ONE (1) zip file name "Lab1.zip".
iii) Upload "Lab1.zip" file to e-learning under [Labs Submission] section

2. Prepare a video demo for debugging process Lab 1(b) and Lab 1(c) – Start Debugging. If your group consists of 3 students, must do Lab 1(a). Each student should explain one debugging process.

Requirement for the video:
- – Video duration is about 10 to 15 minutes
- – You MUST explain verbally on how you execute this Lab 1 in the video (… show your face in recording, please ☺)
- – The video can be recorded in any tools that you're comfortable
  - • Example; you may use any online meeting tool like Zoom, Google Meet, MS Teams to record your lab demo.

3. Video Submission

Demo video can be submitted in two options:

a) as an .mp4 video file (to be uploaded to e-learning) – this is not preferable if the size is big.

OR

b) Upload your video to your YouTube account and share the YouTube video link in the submission option. Set your YouTube video unlisted video (i.e. only person whou has the link can see the video)
- • Upload OR share the YT video link to e-learning under [Lab Submission] section.

Method (b) is the most preferable.