NAME: ODUNBYE LEONARD MOTOSOROLUWA SIGNATURE:

MATRIC NUMBER: 17/ENG02/061

COURSE CODE: COE 206

1. (a) The reason why it won't be a good idea to use numeric addresses when writing instructions that access variables is because the numeric addresses coded in the instructions would have to be updated whenever new variables were inserted before the already existing ones.

(b) The files that are produced by the assembler are
(i) Object File: A machine language translation of the program
(ii) Listing File

2. (a) Assembly language is not portable because an assembly language is tied to 1 specific computer architecture. A program written in one assembly language would need to be completely rewritten for it to run on another type of machine. Portability is one of the advantages that higher level languages have over low-level languages like Assembly Language.

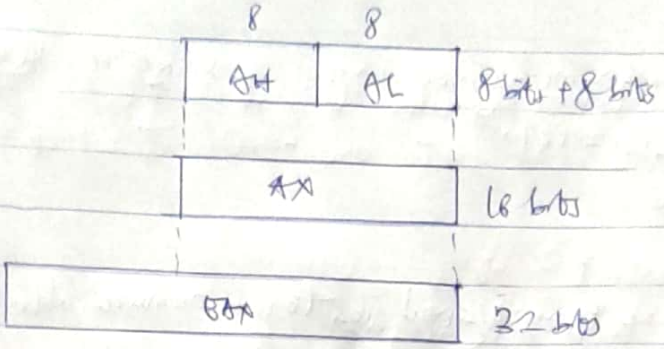
(b) No, the reason is that each assembly language is primarily written and built for a specific set/family of processors or computers.

(c)

32-bit	16-bit
ESI	SI
EDI	DI
EBP	BP
ESP	SP

(i) ESI and EDI: These registers are used for high speed memory transfer instructions. They are also known as Extended Source Index (ESI) and Extended Destination Index (EDI)
(ii) EBP: This register is used by high-level languages to reference

(c)



(2) (a) Segmentation is achieved in assembly language through the division of the system memory into groups of independent segments referenced by pointers located in the segment registers. Each segment is used to contain a specific type of data. In the protected mode of memory management, there are 2 segmentation models:

(i) Flat segmentation model

(ii) Multi-segment model

(b) (i) main PROC - The "PROC" directive is used to identify the beginning of a procedure in a program. "main" is the name of the procedure in this program

(ii) MOV AX, 47104 - The "MOV" instruction/keyword is used to move or copy the integer, 47104 into the AX register.

(iii) ADD EAX, 1270 - The "ADD" keyword is used to add the octal integer, 1270 to the EAX register.

(iv) MOV DS, AX - The integer in the "AX" register is copied or stored in the DS register.

(v) main ENDP - The "ENDP" directive is used to mark the end of the main procedure.

(c) (i) value1 BYTE 6Ah - This instruction stores the 8-bit hexadecimal integer, 6Ah in the variable, value1.

(ii) value2 DWORD ? - Stores an unsigned, unidentified 32-bit integer in the variable, value2.

(iii) value3 SBYTE -10, -20, -30, -40, -50 - a sequence of define signed bytes are

stored is the value 3 integer. The value -10 is at offset 0, -20 is at offset 1, -30 is at offset 2, -40 is at offset 3 and -50 is at offset 4.

(4)

```
TITLE Subtract 3 integers (Sub.asm)
; This program subtracts 16-bit integers.
INCLUDE Irvine32.inc
.data
value1 WORD 3000h
value2 WORD 5000h
value3 WORD 1000h
.code
main PROC

    mov ax, value1;
    sub ax, value2;
    sub ax, value3;
    call DumpRegs

    exit
main ENDP
END main
```