Alexis John Abalkasanga
17/ENG02/001
COE306 TEST

Question 1

(a) Why would it not be a good idea to use numeric addresses when writing instructions that access variables?

Using numeric addresses ties you to specific locations in memory. Normally applications can be loaded anywhere in memory, so it won't work if it's loaded in a different place than you assumed when you programmed it. Also, because the addresses coded in the instructions would have to be updated whenever new variables were inserted before existing ones.

(b). What types of files are produced by the assembler?

They are; Object .OBJ & listing .LST files & a Source file.

Question 2

(a) Briefly explain the concept of portability as it applies to programming languages

Portability is a characteristic attributed to a computer program if it can be used in an operating system other than the one in which it was created without requiring major reworks.

(b) Is the assembly language for x86 processors the same as those for computer systems such as the AMD or Motorola 68x00? Give reason

The assembly language for x86 processors are not the same for other computer systems such as AMD or Motorolla 68x00 because each assembly has its own specific processor or computer

c) The general purpose register are used mainly for arithmetic & data movement, explain the lower families of EAX register with aid of a diagra

32bit registers

16bit registers

31	16	15 8	7 0	
EAX		AH	AL	AX Accumulator
EBX		BH	BL	BX Base
ECX		CH	CL	CX Counter
EDX		DH	DL	DX Data

AX : is the primary accumulator
BX : is known as the base register, used in indexed addressing
CX : is known as the Count register, registers store the loop Count iterative operations.
DX : is known as the data register, also used in input/output opera

Question 3

a) Explain how Segmentation is achieved in assembly language.
A segmented memory model divides the system memory m groups of independent segments referenced by pointers located in the Segment registers

b) Main Proc - This identifies the beginning of the code
- MOV AX, 4710H - This tells the program to move '4710H' in to the register Ax
- ADD EAX, 127Q - This tells the program to add 127 C in base8) to the value already existing in EAX register
- MOV DS, AX - This tells the program to move the value in AX in to DS
- Main ENDP - This is the exit statements that calls a predefined MS-Windows function that halts the program

c) Value 1 Byte 60h - This tells the system to store byte 60h under value 1 label & It's an unsigned byte
Value ↓ Byte ↓ Radix
Label Directive Initializer

d) Value 2 DWORD ? - This is an uninitialized variable & it's value

can be assigned at runtime.

Question 4

```
This program Subtracts 3 16-bit integers
INCLUDE Irvine32.inc
.data
Val1 WORD 6000h
Val2 WORD 3000h
Val3 WORD 2000h
final val ?
.Code
Main PROC
mov eax, val1
Sub eax, val2
Sub eax, val3
Mov final val,
Dump Regs
exit
Main ENDP
/* Written by Alexis John, 17/ENG02/001 */
END main
```