# Exam 1

## Spring 2011

## CSCI 2160

**Name: _____**
**Last First Middle**

CSCI 2160 Part 1: 82 points

Complete the below table:

1. (6pts)

| Character | Hollerith Code (punches) | EBCDIC code in **Binary** | AScii Code in **hex** |
|---|---|---|---|
| G | 12-7 | 11000111 | 47 |
| P | 11-7 | 11010111 | 50 |
| W | 0-6 | 11100110 | 57 |
| 7 | 7 | 11110111 | 37 |

**For Questions 2,3,4 use the portion of the data segment given in #2**

2. (8pts -each answer worth 1 pt) Given the below portion of a data segment of a program, determine the values for the location counter and write the **actual** bytes in memory.

```
02DC  X     word  27,-1,35    __1B00FFFF2300_____________________________

02E2__      Y     dword 2 dup(8)    08000000 08000000_________________

02EA__      T     qword 13          _0D00000000000000_________________

02F2__      strPr byte  “AB123”,6    414231323306_____________________________________

02F8____
```

3. (5 pts) Suppose you are given the below instructions. Write the words “correct” if the statement will assemble and if it will not, write on the line what is wrong with this instruction.

```
a)   MOV    Al, -180    __INVALID – largest negative value is -128____________

b)   ADD    BX, Y       ___INVALID sinzes don’t match___________

c)   MOV    X, X+4      _-INVALID = cannot do memory-to-memory_____________

d) MOV      eax, Y      _______VALID_____________________________________________

e)   ADD    AL,strPr    ______VALID_____________________________________
```

4. (6 pts) Assuming that the following instructions have occurred, give the value that will be in the AX register after the instruction is executed. Assume for all questions that the given two instructions had just been executed . Be sure to reference #2 where necessary and note the question is asking what is in **AX** register.

```
        MOV  AX, 02DCh
        MOV  EBX, offset,strPtr   EBX: 000002F2

       a)  MOV    AX,[EBX]    _4241_______

       b)  MOV    AX,[EBX+3]  __3332_______

       c)  MOV    AX, X[1]    ____FF00_______
```

5. (5) Determine the result in the AL register after each of the below operations. Also determine the settings of the OF, CF, SF and ZF in each case

```
        Add   AL,T  ; assume this is the instruction to be executed

                          CF          OF          SF          ZF
a) AL :  11110110
   T:    11010001

   AL:   __________     ___         ___         ___         ___


b) AL :  10010000
   T:    10110001

   AL:   __________     ___         ___         ___         ___
```

6. (10 pts) Given that ints are 4 bytes, shorts are 2 bytes, write the assembly code generated by the below statements (i.e. what does the data segment look like?)

```
short sX;                          __sX  word   ?_____________________

int   iV;                          _iV  dword   ?_____________________

int   iZ[3];                       _iZ   dword  3 dup (?)_____________________

short[]  sVal = {-1,15,-2);        _sVal word  -1,15.2_____________________

byte [] bT = {6,6,6,6,6,6,6};      bt byte 7 dup (6)_____________________
```

7. (12) Given the above definitions, write the code generated in assembly by each of the Java statements below (Note: in Java [1] references the 2nd element in the array, NOT the first):

```
sX  =  sVal + iZ[1];

              mov   ax, sVal
              cwde                ; eax = sVal
              Add   eax, iZ[4]
```

| iZ[0] | iZ[1] | iZ[2] | iZ[3] | iZ[4] | iZ[5] | iZ[6] | iZ[7] | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | |

```
iV =  bT[2];

mov al, bT[2];
cbw                 ;ax = bT[2]
cwde                ; eax = bT2]
mov iV, eax



sX = iV;
mov eax, iV
mov  sX, ax
```

8.. Given that sX contains `0EC4 h` and that bVal contains `FFh`.

a) (3) Write the code which will compute `sX += bVal;` `sX = sX + bVal;`

```
mov al, bVal
cbw                ;ax = bval
add   sX, ax
```

b) (1) What will be the result of that addition (what will be in sX)?

al: FF
ax: FFFF
sX: 0EC4
FFFF
0EC3

9. (2) What are the index registers? _____`ESI,EDI`

Use the below assembly listing of **exam1.lst** to answer the next set of questions.

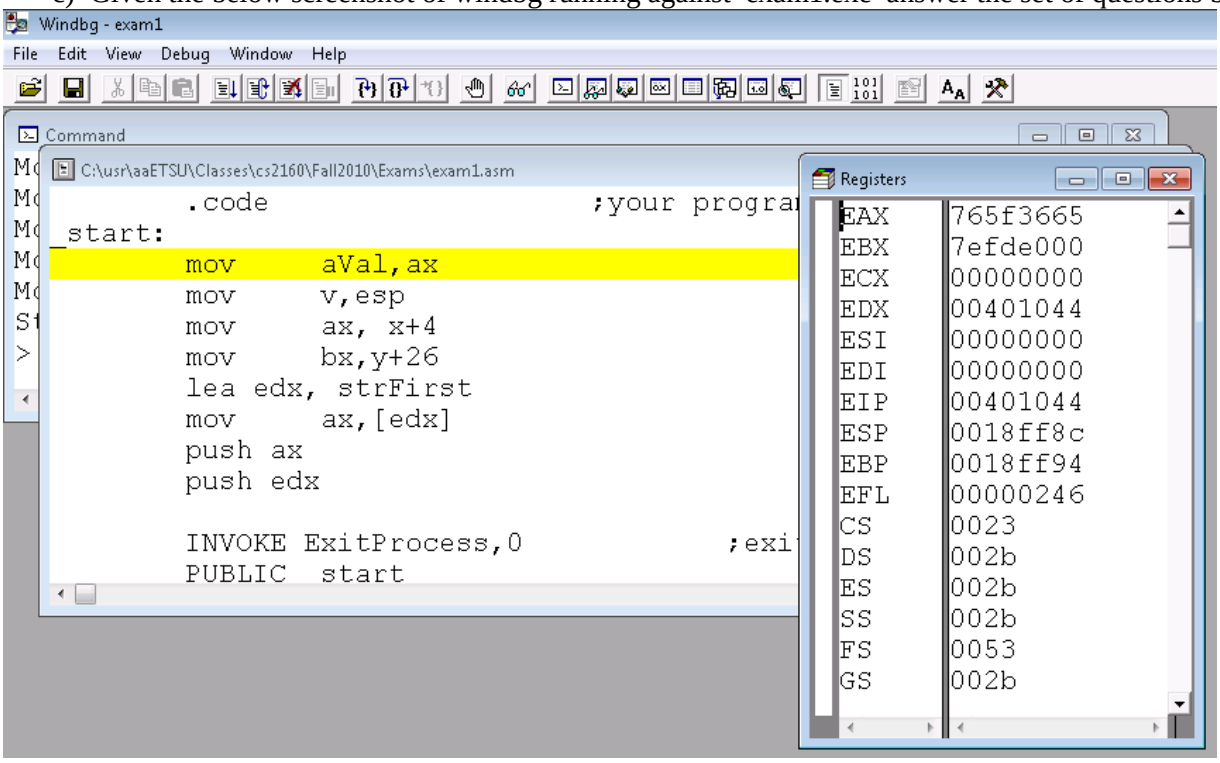
```
                        .model flat
                        .stack 100h

                        ExitProcess PROTO Near32 stdcall, dwExitCode:dword  ; capital
00000000                    .data
00000000  00000003 [        x   word    3 dup (19,22,47)
      0013 0016
      002F
     ]
00000012  004B 0052             word    75,82
00000016  00000004 [        y   word    4 dup (-1,13,-25)
      FFFF 000D
      FFE7
     ]
0000002E  0039 0017             word    57,23
00000032  00000004 [        z   word    4 dup(43)
      002B
     ]
0000003A 21 0F 13       t   byte    33,15,19
0000003D 0000007F       u   dword   127
00000041 FFFFFF80       v   dword   -128
00000045 0000           aVal    word    ?       ;used for preserving ax
00000047 0000           bVal    word    ?       ;used for preserving bx
00000049  00000005 [        dVal    dword   5 dup (54)
      00000036
     ]

0000005D 48 41 4E 4B 00     strFirst byte  "HANK",0
00000062 53 4E 4F 57 00     strLast  byte    "SNOW",0
00000067 0A 0D 00       CRLF    byte  10,13,0

00000000                    .code           ;your program instructions go after this
00000000                _start:
```

10. a) (4) What is the size of the

a) (1 pt) stack in hex ? __100h_____

b)(2 pts) data segment in hex? _6A_____

c) Given the below screenshot of windbg running against exam1.exe answer the set of questions below:




(1 pt) What is the starting address of the code segment? ___00401044

(1 pt) What is the address of the first byte beyond the TOP of the stack? 0018ff8c

(1 pt) What is the address of the very FIRST byte at the BOTTOM of the stack? _0018fe8C_____

(2 pts) Given the windbg screen above, and the copy of the assembly listing, **exam1.lst** above, explain in detail the steps you would take using windbg to display the data segment.

11. Answer each of the following: (1 pt each)

a) What is the largest unsigned value that will fit in a byte? 255_____

b) What is the largest signed value that will fit in a byte? -128 127_____

c) Given that 11010010 is the value in a byte ; D2h 208 +2

i) what is the unsigned decimal value? _210_

ii) What is the signed decimal value? ;2D+1 -46

12.(2) Given the below portion of a code segment, what is the size of the code segment? _____

```
00000000  68 00000000 R        push    offset  rootnode    ;push the address
00000005  E8 00000050          call inorder
                         newline
0000000A  66| 50         2     push    ax      ;preserve ax
0000000C  B0 0D          2     mov al,0Dh      ; display the character
0000000E  50             2     push    eax     ; using putchar
0000000F  E8 00000000 E     2      call    putchar     ;
00000014  83 C4 04       2     add esp,4       ;restore the stack pointer
00000017  66| 58         2     pop ax      ;restore the preserved ax
00000019  66| 50         2     push    ax      ;preserve ax
0000001B  B0 0A          2     mov al,0Ah      ; display the character
0000001D  50             2     push    eax     ; using putchar
0000001E  E8 00000000 E     2      call    putchar     ;
00000023  83 C4 04       2     add esp,4       ;restore the stack pointer
00000026  66| 58         2     pop ax      ;restore the preserved ax
00000028  E8 000000BD          call preorder
                         newline
0000002D  66| 50         2     push    ax      ;preserve ax
0000002F  B0 0D          2     mov al,0Dh      ; display the character
00000031  50             2     push    eax     ; using putchar
00000032  E8 00000000 E     2      call    putchar     ;
00000037  83 C4 04       2     add esp,4       ;restore the stack pointer
0000003A  66| 58         2     pop ax      ;restore the preserved ax
0000003C  66| 50         2     push    ax      ;preserve ax
0000003E  B0 0A          2     mov al,0Ah      ; display the character
00000040  50             2     push    eax     ; using putchar
00000041  E8 00000000 E     2      call    putchar     ;
00000046  83 C4 04       2     add esp,4       ;restore the stack pointer
00000049  66| 58         2     pop ax      ;restore the preserved ax
0000004B  E8 00000052          call postorder
00000050  83 C4 04          add esp,4   ;restore the stack pointer

                         INVOKE ExitProcess,0
0000005A                 PUBLIC _start
```

++++++++++++++++++++++++++

13. (2) Why are there 8 bits in a byte (2 reasons)?