Class Exercise 1

1. Make a diagram showing successive bytes of memory like those above to show the memory layout produces by the following data declarations:

```
Letters     DB    'ABC'
Digits      DB    1, 2, 3
Numbers     DW    6767h, 0abbah        ;h indicates hex
More        DB    'e', 10, 'fg'
Hush        DB    5 DUP ('S'), 'H!'
Two3        DB    3 DUP (2, 3, ?)
Recurse     DB    2 DUP ('X', 3 DUP (0))
```

(Hint : the '0' in 0abbah is merely to indicate to the assembler that a number follows.)

2. Given the following data definitions:

```
W           DW          1234
A           DB          23
B           DB          -12
```

which of the following **mov** instructions are illegal and why?

```
mov     W, 74
mov     al, B
mov     74, W
mov     A, 74
mov     W, ah
mov     A, B
```

3. Fill the contents of the specified registers as four hex digits, given the data definitions.

```
AWord       DW      5432h
AByte       DB      9ah
Another     DB      0bch
```

```
mov     ax, 1234h;          ax = 1234h

mov     ax, AWord;          ax = 5432h
mov     ax, 1234h;
mov     ah, AByte;          ax = 9a34h
mov     ah, AByte ;
mov     al, Another;        ax = 9abch
mov     ax, 1234h;
mov     al, ah;             ax = 1212h
mov     ax, 1234h;
mov     ah, 'A';            ax = 4134h      (HINT: 'A' = 65 = 41h)
mov     ax, 1234h;
mov     ax, 'A';            ax = 0041h
mov     ah, 1;
mov     al, 2;              ax = 0102h
```

4. What is displayed by the DOS display string call:

```
mov   dx, OFFSET Msg
mov   ah, 9h          equivalent to _PutStr Msg
int   21h
```

for each of the following definitions of **Msg**? Assume that the cursor is at the left-hand side of the screen before each display. Show the final position of the cursor with an underline.

a)
```
Msg   DB   'All '          All on one line
      DB   'on one'        _
      DB   ' line', 13, 10, '$'
```

b)
```
Msg   DB   'One $', 13, 10, 'Two', 13, 10, 'Three', 13, 10, '$'
```
One _

c)
```
Msg   DB   4 DUP (5 DUP ('*'), 13, 10), '$'
```

d)
```
Msg   DB   10 DUP ('*'), 13, 10
      DB   3 DUP ('*', 8 DUP (' '),'*',13, 10)
      DB   10 DUP ('*'), 13, 10, '$'
```
```
**********
*        *
*        *
*        *
**********
_
```

5. Write data definitions for **Msg** so that the DOS display String Call:

```
mov   dx, OFFSET Msg
mov   ah, 9h          equivalent to _PutStr Msg
int   21h
```

will produce the following output:

a)
```
Eat
Neat
```
```
Msg  DB   'Eat',13,10,'Neat$'
```

b)
```
down
   down
      down
```
```
Msg  DB   'down',13,10,'  down',13,10,'    down$'
```

c)
```
Manchester United: 2 Manchester United: 2
Middlesborough: 3 Middlesborough: 3 Middlesborough: 3
```
```
Msg DB 2 DUP ('Manchester United: 2 '),10,13
    DB 3 DUP ('Middlesborough: 3 '),'$'
```