Bitwise Operations (Part 3)
§6.2, §7.2

Administrivia

▸ **Exam 2** Wednesday, November 5

  ▸ Make-up exams must be scheduled **before** the exam is given in class; no make-ups afterward

▸ **Homework 5** out due Friday at 11 a.m.

▸ Reading:

  ▸ §6.2 Boolean and Comparison Instructions

  ▸ §7.2 Shift and Rotate Instructions

    ▸ Pay attention to *rotations* (not covered in lecture)

    ▸ Be able to fill out the instruction template (like we've been doing in class) for ROL, ROR
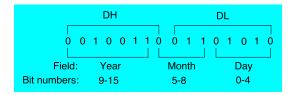
  ▸ §7.3 Shift and Rotate Applications

▸ SHL, SHR, SAL, SAR instructions

Application: Isolating a Bit String

▸ The MS-DOS file date field packs the year, month, and day into 16 bits:

	DH		DL	
	0 0 1 0 0 1 1 0	0 1 1 0 1 0 1 0		
Field:	Year	Month	Day	
Bit numbers:	9-15	5-8	0-4	

▸ We want to "extract" the month field and store its value in AL, so AL = 00000011b

```
mov ax,dx          ; Copy DX into AX, so AX = 0010011001101010
shr ax,5           ; Shift right 5 bits, so   AX = 0000000100110011
and al,00001111b   ; Clear bits 4–7 in AL         AL = 00000011
mov month,al       ; save in month variable
```

Activity 13 #4

Topics Covered in Notes:

▸ Example: Converting binary numbers to strings

```
▸ push OFFSET buffer
  push 42
  call BinToStr ; Now buffer contains the null-terminated string "00000000000000000000000000101010"
```

```
; (STDCALL) Stores a null-terminated string with the binary representation of a
;              32-bit unsigned integer value.
; Receives:   [ebp+8]    DWORD value to convert
;             [ebp+12]   Pointer to buffer to store string (≥ 33 bytes)
; Returns:    (none)
BinToStr PROC
        TODO: Fill this in

BinToStr ENDP
```

Topics Covered in Notes:

▸ Converting binary numbers to strings

▸ Multiplication by 2

▸ Multiplication by 2^n

▸ Division by 2^n, rounding toward $-\infty$