Week 9

* C
* **divide by 2**
  + Bitwise
    - AND = &
    - OR = |
  + Bit pattern of a 32 bit float is as follows:
    - SEEE EEEE EMMM MMMM MMMM MMMM MMMM MMMM
  + Exponent mask is 0x7F80 0000
  + Mantissa mask is 0x007F FFFF
  + Sign mask is 0x8000 0000 (Hidden bit)
  + Highest stored mantissa bit = 0x0040 0000
  + int expField = bitPattern & EXP\_MASK
  + int exponent = exponent field over 23 bits
  + int mantissa = bitPattern & MANT\_MASK
  + int signField = bitPattern & SIGN\_MASK
  + Three cases
    - exp == 0.00
      * //denormalized
      * mantissa >>= 1;
    - exp == 0x01
      * //becoming denormalized
      * exp--; //This makes the mantissa 0x00 because we know it is 0x01
      * mantissa >>= 1;
      * mantissa = mantissa | HIGHEST\_STORED\_MANT\_BIT
    - else
      * //normal
      * exp--;
    - now we need to return all of the different parts together to return the new float
      * To do so, we bitwise or all of the parts together
      * bitPattern = signField | (exp << 23) | mantissa
      * return (\*(float\*)&bitPattern) //Give me the address of bitPattern, dont think of that as the address of an int pointer, think of it as a float pointer, and finally (\*), give me that damn pointer!
* Assembly
  + If we’re returning an int, we place it into EAX by convention