High level languages translates statements into the machine language of that computer, and to calls to subroutines within the OS.

I. Reasons to know about Machine dependent language.
   A. Understand limitations of specific machines or OS.
      int X = 28502, Y = 12344, Sum;
      Sum = X + Y;
      Sum is -24689 in a 16 bit machine because of overflow.
   B. Portability issues
      1. Big Endian (MSB has lower address) Motorola and IBM vs. Little Endian (LSB has lower address) Intel
      C. Access to special hardware, e.g. in embedded applications.

II. Reasons not to use ML
   A. Inconvenient to write and hard to read.

III. CISC vs RISC
   A. CISC
      1. Small code does complex things, makes programs smaller, but longer average instruction time
   B. RISC
      1. More code per task, but each instruction more uniform and therefore clock can be faster.

Information Representation-- How do we represent data?

I. Bits and bytes
   A. bits = binary digit.
   B. Start counting from right, at bit 0
   C. Base N to Decimal Conversion
      1. Multiply each digit by its weight and adding up the result.
         {Digits are processed left to right }
         Value = 0;
         repeat
         value = value * N + next_digit
         until (no digits remain)
   D. Decimal to Base N Conversion
      1. Horners Algorithm
         {Digits are produced right to left, i.e. LSD first}
         Value = value to be converted.
         repeat
         next digit of result = value mod N;
         value = value div n;
         until value = 0;
   E. Binary/Octal/Hexadecimal Conversion
      1. Remember to replace hex 0 with four zeroes.

II. Integer representation
   A. Unsigned integers
   B. Signed integers
      1. Signed magnitude
      2. 2's complement.
         a) digit odometer
         b) To find the n-bit 2's complement of -x
            (1) Find the n-bit base 2 rep of +x
            (2) In the result of (1) replace 0's by 1s and 1s by 0s;
               (a) In hex F<>0, E<>1, D<>2 etc.
            (3) Add 1 to the result of (b) and ignore carry of MSB
         c) -x = 2^n - x
         d) Properties of 2's complement
            (1) Range is -2^(n-1) to 2^(n-1) -1
(2) \( (n-1) = 100...000 \) and \( 2^{(n-1)} - 1 = 011....111 \)
(3) All non-negative numbers have a 0 in Bit \( n - 1 \);  

3. Addition  
   a) Unsigned and 2's complement the same.

4. Subtraction  
   a) Unsigned and 2's complement the same.  
   b) In practice computers negate second operand and then add.

C. Overflow - when result can't represented in the number of bits used. Two flags, one for each type.  
   1. Unsigned - when carry bit is affected.  
   2. Addition carry out of MSB  
   3. Subtraction borrow into MSB  
   4. Signed - when sign of result doesn't match first operand's sign  
   5. Addition +ve + +ve -> -ve or -ve + -ve -> +ve. No overflow possible when different signs.  

III. Chars  
   A. Seven bits with 0 in bit 7

IV. Real Numbers - actually Floating Point  
   A. Scientific notation  
   B. Converting decimal fraction to base N  
      1. { Digits are produced left to right, ie first digit is MSB}  
         Value = fraction to be converted.  
         repeat  
         Value = Value * N;  
         Next digit of result = integer part of value  
         Value = Fractional part of Value  
         until (value = 0) or (the desired number of digits are produced.

C. Book representation  
   1. 24 bit representation: 8 bit exponent 1 bit sign 15 bit mantissa  
      a) Exponent is excess 128, i.e. add 128 to exponent. 0 is zero value.  
      b) Sign bit, 1 = negative  
      c) Mantissa is normalized with implicit 1.

D. IEEE (Institute of Electrical and Electronic Engineers) Representation  
   1. Exponent bits with bias of 127 (add 127 to exponent), 0 is for 0, and 255 is for infinity  
   2. 32 bits: 1 sign, 8 exponent, 23 mantissa  
   3. 64 bits: 1 sign, 11 exponent (add 1023 to exponent), 52 mantissa.