CISC124 – Today’s Topics

- Quiz 2
- Numeric representations
- Internal representation of numbers
  - Integers (two’s complement)
  - Real numbers (IEEE 754 floating point)

Quiz 2

Location and Hours:
- Quiz 2 will be written in Jeffery 155, at the beginning of your lab session
- Mon Feb 25, 9:30 am – 10:30 am
- Mon Feb 25, 2:30 pm – 3:30 pm
- Tue Feb 26, 8:30 am – 9:30 am
- Wed Feb 27, 2:30 pm – 3:30 pm

Form for:
- One coding question
- A set of multiple choice questions

Quiz 2

Topics (Everything covered from Jan 29 until Feb 13):
- 2D arrays
- Ragged arrays
- Aliasing. System properties
- Exception throwing and handling. Exception classes
- Try-catch-finally statement
- Wrapper classes
- Foundational classes: Math, String, StringTokenizer
- Method overloading
- File I/O: text and binary files
- Software qualities
- Classes, objects, encapsulation

Numeric representations

- Representation of a numeric value (counting or measuring intuition) using a system of digits and a method to combine those digits using powers of a base number
- Digits (coefficients) & Base (radix)
  - Decimal System \(0, 1, 2, 3, 4, 5, 6, 7, 8, 9\) (Base \(r = 10\))
  - Hexadecimal System \(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\) (Base \(r = 16\))
  - Octal System \(0, 1, 2, 3, 4, 5, 6, 7\) (Base \(r = 8\))
  - Binary \(0, 1\) (Base \(r = 2\))

- Method (Represent \(n\)-digit number by a polynomial of degree \((n - 1)\))
  - (Integer) \(N = d_{n-1} \times r^{n-1} + \ldots + d_0 \times r^0\)
  - (decimal) \(D = d_{n-1} \times r^{n-1} + \ldots + d_0 \times r^0 + d_{-1} \times r^{-1} + \ldots + d_{-m} \times r^{-m}\)

Numeric conversions

- Any other number representation to decimal \(\rightarrow\) just replace the digits and radix by their decimal representations and do decimal math.
- Decimal number representation to others (hexadecimal, octal, binary) requires simple algorithm:
  1. Divide decimal number by new base (integer division) \(\rightarrow\) remainder is last digit of new number representation
  2. Divide quotient of previous division by new base \(\rightarrow\) remainder is second last digit of new number representation
  3. Perform steps 1 to 2 until quotient is 0 \(\rightarrow\) remainder is left-most digit of new number representation
- Between number representations whose bases are powers of 2 (hexadecimal (16), octal (8), binary (2)) just complete a 1 or 2 step algorithm:
  - Convert each digit to its binary representation
  - Regroup them in groups of the following sizes: 4 for hexadecimal, 3 for octal, 1 for binary
Internal representation

- **Integers**

  - Two's complement encoding for positive and negative integers
  - Fixed number of bits to represent an integer: 8, 16, 32, 64
  - 1 sign (left-most) bit: 0 if positive integer, 1 if negative integer
  - Representation of positive integers: 0 sign bit plus unsigned binary representation of integer
  - Representation of negative integers: one is added to inverted representation of positive integer (0's are flipped to 1's and vice versa)
  - Addition and subtraction are reduced to just unsigned binary addition. Easy to convert results back to decimal representation

**Example:**

- Integer 45 in an 8-bit representation:
  - Positive: 00101101
  - Negative: 00011000

  
  \[
  00011000 + 11101000 \\
  \underline{\text{-------------}} \\
  00010101
  \]

  - Two's complement: 00010101 = 21 (decimal)

**Overflow problem:**

- Arithmetic operations can produce results outside the representation range

**Example:**

- Positive integer 105 in an 8-bit representation:
  - Positive integer: 00101101
  - Positive integer 34 in an 8-bit representation:
  - 105 + 34:
    - Value: 01001111
    - Wrong value: 79

**Real Numbers**

- Java follows the IEEE 754 Standard:
  - float (32 bits, single precision), double (64 bits, double precision)
  - Storage format for a java float consists of:
    - 1 sign bit
    - 8 bits exponent field
    - 23 bits mantissa field

**Example:**

- Value: 0 01111100 01000000000000000000000
  - Value = \((-1)^{0}\times 2^{-127} \times (1 + \sum_{b=0}^{23} b(23-b) \times 2^b)\)

- Value = 0.15625

**Internal representation boundaries for a float:**

- Max. Value = \((-1)^9 \times 2^{254-127} \times (1 + \sum_{i=1}^{23} b(23-i) \times 2^i)\)
  - is constant Float.MAX_VALUE = 3.4028235E38

- Smallest Value = \((-1)^9 \times 2^{-127} \times (1)\)
  - is constant Float.MIN_NORMAL = 1.17549435E-38

- Special values:
  - 0 = \((-1)^0 \times 2^0 \times (1)\)
  - +Infinity = \((-1)^0 \times 2^{255} \times (1)\)
  - -Infinity = \((-1)^0 \times 2^{255} \times (1)\)
  - NaN = \((-1)^0 \times 2^{255} \times (1 + \sum_{i=1}^{23} b(23-i) \times 2^i)\)