



# COMPUTER SCIENCE COMPETITION - JAVA TOPIC LIST 2018-2019

**IMPORTANT NOTES:** Java is the official programming language for UIL Computer Science. Contest content for 2018-2019 will conform to the Java Platform Standard Edition, Version 8. This list is intended as a guideline and is not all-inclusive. Knowledge of basic Java concepts is assumed. Visit the UIL web site at [uiltexas.org/academics/computer-science](http://uiltexas.org/academics/computer-science) for a list of Java resources and other important contest information.

Base Conversions and Arithmetic
User-Defined Classes (constructors, methods, instance variables, private vs. public, overloading, overriding, final local variables, static final class variables, static methods, static non-final variables)
Constructors and initialization of static variables, default initialization of instance variables
Concepts of inheritance, abstract classes, interfaces and polymorphism
null, this, super, super.method(args), super(args), this.var, this.method(args), this(args), instanceof
Conversion to supertypes and (Subtype) casts
Comparison of reference types (equals(), ==, !=, Comparable.compareTo())
Primitive types (int, double, boolean, short, long, byte, char, float), casting of primitives
Arrays, including arrays of arrays and initialization of named arrays
Arithmetic operators (+, -, *, /, %, ++, --) and string concatenation
Using the values of ++, -- expressions in other expressions
Assignment operators (=, +=, -=, *=, /=, %=)
Boolean expressions and operators (==, !=, <, <=, >, >=, &&,   , !, &,  , ^) including short-circuit evaluation
Bitwise operators (<<, >>, &, ~,  , ^, >>=, <<=, !=, &=,  =, ^=)
Branching (if, if/else, ?:, switch, break)
Looping (while, for, enhanced for, return, do/while, break, continue)
System.out.print(), println(), printf(), %f, %d, %s, escape sequences \n, \\, \"
Parsing (String.split(), Integer.parseInt(), Double.parseDouble())
Pattern class, Regex (. + * \d \D \s \S \w \W [abc] [^abc] [a-zA-Z]), matches
Java Standard Library (String, Integer, Double, Character, Math, Object, Comparable, Scanner, Random, Arrays) See supplemental class reference list.
Generic collections (Collection, List, Set, Map, Stack, Queue, PriorityQueue, ArrayList, LinkedList, HashSet, TreeSet, HashMap, TreeMap) See supplemental class reference list.
Arrays.sort() and Collections.sort()
Recursion, Dynamic Programming (dynamic programming applies only in the programming portion of the contest, not the written tests)
Stacks, Queues, Binary Trees, Linked Lists, Heaps, Hash Tables, Priority Queues, Graphs
Sorts (Selection, Insertion, Mergesort, Quicksort) and Searches (Sequential, Binary) – same canonicals as AP
Analysis of algorithms: informal comparison of running times, exact calculation of statement execution counts, Big-O notation, best case / worst case / average case time and space analysis
Digital Electronics – symbolic representation of Boolean expressions using logic gates NOT, AND, XOR, OR, NAND, NOR, NXOR
Two's complement binary representation of negative 8-bit integers – conversion both ways between base 10 and base 2
Polish notation – representation, analysis, and conversion of simple infix, prefix, and postfix expressions
Boolean simplification using generic notation ( $A * B, A + B, A \oplus B, \overline{A}, \overline{A * B}, \overline{A + B}, \overline{A \oplus B}$ – using truth tables and Boolean Identities to analyze and simplify Boolean expressions (see list of identities on UIL CS website)

# UIL Computer Science Topic List 2018-2019

## Written Test First 15 Questions

1. **Number base concepts**, arithmetic, conversion
2. **Simple literal math expression** with mixed operations
3. **Simple output** involving print, println, printf (%d, %f, %s, \", \\, \n)
4. **String class methods**
5. **Simple Boolean logic** (AND, OR, XOR, NOT) - Java based
6. **Math class methods** (no advanced topics like trig - save that for later in the test)
7. **Simple variable expression** with mixed operations
8. **Conditionals** (if, if/else, switch - not ternary)
9. **Simple output loop**
10. **1D primitive array**, basic concepts
11. **Input concepts** – use of Scanner and File classes
12. **Accumulation loop** – summation, product accumulation, etc.
13. **Order of operations** (beyond just the math expressions - testing knowledge of the full Java spectrum of order of precedence)
14. **Java specific data type concepts**, memory size, max and min limits, wrap around, complements (~)
15. **ArrayList** – generics only

## Topics for last 5 written test questions (40 question test)

*(three multiple choice, two free response, one discrete answer each)*

- **Boolean algebra concepts** – truth table analysis, use of Boolean identities for simplifying expressions.
- **Digital Electronics** - interpretation of symbols, sketching DE circuits from expressions
- **Polish notation** - representation, analysis, and conversion of simple infix, prefix, and postfix expressions
- **2s complement binary negative representation** (limit 8 bits)
- **Graph theory** - simple paths, cycles, analysis
- **Data structure theory** - stacks, queues, priority queues, binary trees (heaps, search, expression, generic)

## Programming – the first two programs of any packet will be as follows:

- A program that requires only output and will not read data from a file
- A program that will read from a data file, but will be very basic in nature, with little or no significant computation involved