Theoretical Questions for CSc 600 Final Exam

Scheme

- Concepts of functional programming
- Data types
- Quote and symbols
- Arithmetic operators and library functions
- Recognizers and tests
- Equivalence predicates
- Logical values and operators
- Characters and strings
- Pairs
- Lists and list formats
- Proper and improper lists
- List operators (car / cdr / cadr / caddr / cadddr)
- Vectors
- Definition of objects
- Bindings (define/set/let)
- Lambda expressions (functions)
- Evaluation of expression (eval)
- Map
- Apply
- Control structures (begin / if / cond / case / do / exit)
- I/O operators
- *the-non-printing-object*
- Pretty print
- Functions with arbitrary number of arguments
- Compound functions
- First-class objects
- Scope
- File I/O (inport/outport)

Ruby

- Concept of multi-paradigm languages
- Ruby as a multi-paradigm language
- Types of language processors
- Interpreters, virtual machines, compilers
- Performance of language processors
- Ruby interpreter and VM
- Object oriented concepts in Ruby
- Ruby vs. other HLL
- Ruby execution modes/environments
- Numeric data types
- Class belonging of objects
- Constants (syntax and examples)
- Ranges and their use
- Dynamic typing of objects
- Classification of variables in Ruby
- Parallel assignment
- Use of blocks and block expression
- Classification of Ruby control structures
- Conditional statements in Ruby
- Loops in Ruby
- Iterators in Ruby
- Array generators in Ruby
- Modifiers of control flow
- Conditional expressions
- Short-circuit evaluation in Ruby
- Syntax of specific control statement
- If statements
- Unless statement
- Modifiers and their use
- Method and operator precedence
- Case statement
- While and until loops
- For loop and each iterator
- The concept of hash objects
- Loop with implicit counter
- *Upto* and *downto* iterations
- Iterator step
- Iterators collect, select, reject
- Infinite loop and its use
- Break, redo, next, and retry
- BEGIN – END
- The Time class and its use

- Convertors to_a, to_s, to_f
- Text files and their processing
- Dir and File classes
- Open and close of files
- Methods: definitions and examples
- ri – role and use
- Default values of method arguments
- Return values types in methods
- Using methods for side effects
- Arbitrary number of arguments
- Expanding arrays in method calls
- Splat operator
- Array class and array methods
- Intersection and union of arrays
- Repetition (arrays)
- Concatenation of arrays
- Difference of arrays
- Append, include, delete, clear
- Array reverse
- Kernel method p for printing
- Tail recursion in Ruby
- Classes – syntax and examples
- The concept of pseudovariable
- Getters, setters, and accessors in Ruby
- Class variables
- Class methods
- Inheritance in Ruby
- Command line arguments
- Rational numbers and methods
- Complex numbers and methods
- Matrix operations in Ruby

Interpreters

- Language design strategies
- Data types, and arithmetic
- Control structures
- Input-output statements
- General purpose statements
- Domain-specific statements
- Internal form of instructions
- Scanning, parsing, and semantic analysis
- Internal structure of interpreter
- Execution of non-control statement
- Execution of control statements
- Implementation of trace feature
- Examples of translation and execution of programs