

Exam 1 Study Guide

This exam will cover all of the material presented in class. Sample questions are listed below. The sample questions do not necessarily cover all of the content of the exam but they are representative of the types of questions that will be asked.

- Define the terms/phrases: syntax, semantics, precedence, associativity, parse tree, low-level language, high-level language, derivation, compilation, token, interpreter, compiler, virtual machine, logical programming, event-driven programming, concurrent programming, functional programming, imperative programming, operator overloading, scope, weakly/strongly typed, statically typed, dynamically typed, high-order function, data type, primitive data type, coercion, casting, type system, type-error, lexical analyzer, semantic analyzer.
- Give an example of a(n) a) object oriented language b) business-oriented language c) low-level language d) event-driven language etc..
- Give a BNF grammar, a syntax chart, and a regular expression for Java integer literals.
- Give a BNF grammar for expression involving addition, multiplication, division and subtraction. Your grammar must force conventional associativity and precedence rules when parsing an expression.
- Prove that a particular BNF grammar is ambiguous.
- Is BNF able to express the constraint that “all variables must be declared prior to use”? Is BNF able to express the constraint that an integer literal cannot have more than 12 digits? Is BNF able to express the constraint that there must be an expression on the left and the right of the arithmetic-addition operator? Explain your response.
- Give an FSA that accepts the roman numerals I through X.
- Write a scheme function that accepts a list and returns the length of the list. Use only conditionals, car, cons, cdr, and null?.
- Write a scheme function that accepts a list and returns the smallest element of the list. Use only conditionals, car, cons, cdr and null? in addition to relational operators.
- Write a scheme function named “filter” that accepts a predicate and a list. The filter method returns all elements of the input list that satisfy the predicate.
- Define the boolean data type.
- A scheme s-expression is either a symbol, string, number, boolean or list. Describe the evaluation rules for an s-expression.
- Write a scheme function named “listify” that accepts a list of numbers and partitions the elements into sub-lists each of which is in monotonically increasing order. For example

```
(listify '(1 2 3 1 2 3 1 2 1)) => ((1 2 3) (1 2 3) (1 2) (1))  
(listify '(5 4 3 2 1)) => ((5) (4) (3) (2) (1))
```
- What is the result of the following scheme expressions?

```
(map cons '((1 2) (3 4) (5 6)))  
((lambda (x) (- x 3)) 12)  
(cons (cdr '(1 (2 3) 5 6 (7 8))))
```

