TEL AVIV UNIVERSITY
Department of Computer Science
0368.1105 – Extended Introduction to Computer Science
Fall Semester, 2000/2001

Handout for Lecture 1

Fundamental Concepts

Declarative Knowledge vs. Imperative Knowledge

- Declarative Knowledge - “What is true”
  For example:
  $\sqrt{X}$ is the Y such that $Y^2 = X$ and $Y > 0$
- Imperative Knowledge - “How to”
  For example:
  To find an approximation to $\sqrt{X}$
  - make a guess G,
  - improve the guess G by averaging G and X/G,
  - keep improving the guess until it is good enough.

The Elements of Programming

- Primitive Expressions
  - Primitive Procedures
  - Primitive Data
- Means of Combination
  - Procedural Composition
  - Construction of Compound Data
- Means of Abstraction
  - Procedure Definition
  - Simple Data Abstraction
  - Naming

Some Topics

- Black box abstraction.
- Conventional Interfaces
- Generic Operations
- Large Scale Structure and Modularity
- Object Oriented Programming
- Operations on Aggregates

Additional topics

- Meta-Linguistic Abstraction: (chapter 4)
  - Interpretation (Eval/Apply)
  - Language variations
- Computing with Register Machines: (chapter 5, not covered)
  - A simple, machine-like model
  - Compilation

Evaluation of an Expression

- The value of a numeral: number
- The value of a built-in operator: machine instructions to execute
- The value of any name: object associated with that name in the "environment"
- If a special form, do something special.
- To evaluate a combination (other than special form):
  - Evaluate the subexpressions in any order,
  - Apply the procedure that is the value of the leftmost subexpression to the arguments (the values of the other expressions).
- To apply a procedure to the arguments:
  - If procedure is primitive procedure, just do it
  - If procedure is a compound procedure, then evaluate the body of the procedure with the formal parameters replaced by the actual argument values.