

A: Division

Instructional

Date

June 18, 1997

P: Course Content

- 1.2 Primitive data types and expressions
- 1.3 Control Structures
- 1.4 Functions and parameter passing
- 1.5 Arrays
- 1.6 Top-down design review and specs. for assignment #1: procedural programming with

emphasis on control structures, procedures, and arrays

2.1 Strings

2.2 Collections

2.2.1 Lists

2.2.2 Sets

2.2.3 Stacks

3. Implementing Abstractions

3.1 C++ Strings

3.2 Introduction to pointers (domains)

3.3 C++ records (struct)

3.4 Static data structures

emphasis on: cohesion and coupling and using more complicated static data structures

3.5 Design of set primitives

3.6 Recursion

3.6.1 Numerical examples: factorial, Fibonacci, ...

3.6.2 Examples from symbolic (LISP-like) expressions (SExpressions)

3.7 Discussion and specs for assignment #1: functional programming using an existing

module for SExpression and with graphics or other

4 Encapsulation, instantiation, and OO

4.1 Structure (syntax and semantics)

4.2 Examples

4.2.1 Sets implementation and use

4.2.2 Stacks implementation and use

4.2.3 Specs for assignment #2: OO

Q: Method of Instruction

new and area components to the course. The lecture is used to introduce new material, usually via a sequence of theoretical concepts, practical considerations (usually language dependent), and one or more example case studies. The book is to be used as an additional source of problems and examples.

They are marked mostly on results, i.e. correctness of the algorithm. Assignments are the most important. They are marked according to program design, correctness and efficiency of the algorithms, coding style, and completeness of the documentation.

R: Evaluation

Distribution will be given to the student on the first day of classes along with the course outline and necessary policies.

Distribution 2 areas