Computer Science I

CMPE/CSCI 1370 - 01

Composing functions

problem involves multiple subtasks

Composing templates

function accepts multiple parameters

Composing templates

function accepts multiple parameters

- Data definitions + single-parameter templates
- Combine templates
- Simplify cond expression

Simplifying cond expressions

- cond where all answers are the same: eliminate the cond
- multiple conditions need to be true to produce an answer: `and`
- multiple clauses have the same answer: `or`

```
(and (> 7 4) (or (not (> 7 8)) (= 7 5)))
```

A. `true`

B. `false`

C. Error

D. It depends

E. I don't know

Rock paper scissors:

- comparing 2 "hands" to see who wins
- state is composed of 2 pieces: p1 hand and p2 hand

Structs

Compound data: A single value composed of multiple pieces

- Struct definition
- Constructor (we construct *instances*)
- Selectors (we select for *fields*)

Defining structs

- `(`
- `define-struct`
- struct name
- `[` fields `]`
- `)`

What is the correct struct definition for student?

A. `(define student [name id major])`

B. `(define (student name id major))`

C. `(define-struct student [name id major])`

D. `(define-struct student [student-name student-id student-major])`

E. More than one of the above

How do we construct an instance of student?

A. `(make-student "John" 1 "CS")`

B. `(make-student ["John" 1 "CS"])`

C. `(make-student (name "John") (id 1) (major "CS"))`

D. `(make-student 1 2 3)`

E. More than one of the above

Given this instance of a student, how do we select for the student's name?

```
(define s1 (make-student "John" 1 "CS"))
```

A. `(name-student s1)`

B. `(student (name s1))`

C. `(student-name s1)`

D. `(name (student s1))`

E. `(s1[name])`

`posn`s

- constructor `make-posn`
- selectors `posn-x` `posn-y`

What is the output of the following? `(posn-y (make-posn 100 200))`

A. 100

B. 200

C. (make-posn 100 200)

D. (make-posn 200)

E. Error

Which expressions create an instance of Posn with coordinates (8, 12), given `(define posn1 (make-posn 10 12))`?

A. `(make-posn 8 12)`

B. `(make-posn (posn-x posn1) 12)`

C. `(make-posn 8 (+ (posn-x posn1) 2) )`

D. `(make-posn (- (posn-x posn1) 2) (posn-y posn1))`

E. More than one of the above

Structs as world state

Attendance!

http://bit.ly/1370-1rollcall