

# CSE114A lecture 3

## Recap

### Syntax

$$e ::= x \mid \lambda x. e \mid e_1 e_2$$

### Semantics

#### operational semantics —

how programs evaluate step by step

$$(\lambda x. \underline{e_1}) e_2 \longrightarrow_{\beta} e_1 [x := e_2]$$

This means:

" $e_1$ , but with all free occurrences of  $x$  replaced with  $e_2$ , except when doing so would capture variables in  $e_2$ "

$$(\lambda x. (\lambda y. x)) y$$

Naive (incorrect) beta step would give us  $\lambda y. y$

$$(\lambda x. (\lambda y. x)) y$$

$\rightarrow_{\alpha}$

$$(\lambda x. (\lambda z. x)) y$$

$\rightarrow_{\beta}$

$$\lambda z. y$$