CS1101S Midterms

By: cxy

Substitution model

Applicative order reduction

- Evaluates innermost expression(s) before applying outer function(s)

```
add_squares(sq(1 + 2), sq(3 + 4));
add_squares(sq(3), sq(7));
add_squares(3 * 3, 7 * 7);
add_squares(9, 49);
9 + 49;
58;
```

Normal order reduction

- Applies outermost function before evaluating inner expression(s)

Recursive v.s. Iterative

Recursive is a process that builds up a chain of deferred operations.
Iterative is a process that does not "grow" or "shrink", i.e. the deferred operations do not increase as problem size increases.
The

function below is an example of a recursive process.

$$\text{Hi} \approx hello$$

$$\sum_{k=1}^{n} \frac{1}{k} = \text{harmonic sum or smth}$$

$$a = b + c$$
$$d + e = f$$

Scope

Forms of declarations

- Pre-declared names
- Constant declarations
- Parameters of function declarations and lambda expressions
- Function names of function declarations

Scoping rule: a name refers to its *closest surrounding* declaration

Features of this generator

The basic things work: here's some *italics*, some **strong, bold text**, ~~please don't read this~~.
Numbered and unnumbered lists

- work
- I
- guess?

1. Yep,
2. I
3. think
4. they
5. do.

Here's a heading 3

How does a heading 4 look like?

What about... a table?

this	**is**
a	table

We can do `inline code`, and

```
code blocks too!!
```

You can write

inline using `` `$ ... $` ``, and a block

$$\sum_{k=1}^{n} k$$

using ```` ```latex <newline> ... <newline> ``` ````