

## Problem 2

a)

i)

Java, C++, and Python all support some forms of lambda functions

**True**

<https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html>

<https://en.cppreference.com/w/cpp/language/lambda>

<https://docs.python.org/3/tutorial/controlflow.html#lambda-expressions>

ii) Applicative order reduction is at least as efficient as normal order reduction [or vice versa]

**False**

Applicative order reduction can be more efficient than normal order reduction:

*Normal order reduction*

$(\lambda f : f f f) (\lambda x : x) 5$

$((\lambda x : x) 5) ((\lambda x : x) 5) ((\lambda x : x) 5)$

$5 ((\lambda x : x) 5) ((\lambda x : x) 5)$

$5 5 ((\lambda x : x) 5)$

$5 5 5$

*Applicative order reduction*

$(\lambda f : f f f) (\lambda x : x) 5$

$(\lambda f : f f f) 5$

$5 5 5$

Normal order reduction can be more efficient than applicative order reduction

*Normal order reduction*

$(\lambda f : 5) (\lambda x : x) (\lambda x : x) (\lambda x : x)$

$5$

*Applicative order reduction*

$(\lambda f : 5) (\lambda x : x) (\lambda x : x) (\lambda x : x)$

$(\lambda f : 5) (\lambda x : x) (\lambda x : x)$

$(\lambda f : 5) (\lambda x : x)$

$5$

iii)

Reduction of a lambda expression may not terminate

**True**

After one reduction step, the expression

$(\lambda x : x x) (\lambda x : x x) = (\lambda x : x x) (\lambda x : x x)$

Thus, reduction will not terminate.

Marks were not given for this question if you used a named function to achieve self-reference as lambda expressions do not have such capabilities.

Eg.  $f(x) = f(f(x))$

iv)

**True**

B.

Common reduction errors:

$a(\lambda x : x)(\lambda x : x)$  does not reduce to  $a(\lambda x : x)$ . It is in normal form.

c.)

Question well-done overall

Recall logic identities

NOT =  $(\lambda B : B \text{ F T})$

AND =  $(\lambda A B : A B \text{ F})$

OR =  $(\lambda A B : A \text{ T B})$

Find the operator (or combination of operators) that correspond to the truth table,  
Construct lambda expression and reduce