3)
a)
i)

$$\text{(W-ASS.EXP)} \quad \frac{\langle E, s\rangle \rightarrow_e \langle E', s'\rangle}{\langle x := E, s\rangle \rightarrow_c \langle x := E', s'\rangle}$$

$$\text{(W-ASS.NUM)} \quad \frac{}{\langle x := n, s\rangle \rightarrow_c \langle \texttt{skip}, s[x \mapsto n]\rangle}$$

ii)
<y := y + x, s>
-> <y := 1 + x, s>
-> <y := 1 + 3, s>
-> <y := 4, s>
-> <skip, s[y ↦ 4]>

-------------------------- s(y) = 1
<y, s> → <1, s>

<y + x, s> → <1 + x, s>

<y := y + x, s> → <y := 1 + x, s>

b) i)

First prove that <E, s> $\rightarrow_e$ <E', s'> => s = s' (trivial structural induction by small-step semantics and IH).

Lemma:
<E, s> $\rightarrow_e$ <E', s'> => s = s'

We prove this by induction on k in the following statement:
<E, s> $\rightarrow^k$ <n, s'>

Base Case:
<E, s> $\rightarrow_e^0$ <n, s'>
Vacuously true.

Inductive Case:

Assume I.H: $\langle E, s\rangle \rightarrow^{k} \langle n, s'\rangle \Rightarrow s = s'$

To Show:

$\langle E, s\rangle \rightarrow_{e}^{k+1} \langle n, s'\rangle \Rightarrow s = s'$

Assume $\langle E, s\rangle \rightarrow_{e}^{k+1} \langle n, s'\rangle$.

Then $\langle E, s\rangle \rightarrow_{e}^{k+1} \langle n, s'\rangle = \langle E, s\rangle \rightarrow_{e} \langle E', s''\rangle \rightarrow_{e}^{k} \langle n, s'\rangle$

So by lemma s = s'', and then by I.H s'' = s'. Hence s = s'.

ii) As there is only one way to evaluate an expression E (i.e. there is no choice on which term to evaluate first), thus determinacy holds. Therefore E1 = E2 holds. By the results shown in b.i, evaluating expression E has no effect on state. Therefore s = s1 = s2 holds.

Inductively:

See (Lecture 3 page 24) for a very similar proof.

c)

i) It does not hold for b.i as the command C in the expression `do C return E` can be an assignment (from a.i) and thus change the state, thereby allowing a contradiction to occur with the statement of b.i. which is that evaluating expression E does not change state.

However, as determinacy holds even for the extended set of expressions (i.e. there is no choice in which to evaluate first), then E1 = E2 still holds. We can then prove s1 = s2 also holds by induction on the definition of E.

ii)

From notes: (do x := (x + 1) return x) + (do x := (x × 2) return x)

But this is assuming multiplication is defined, which idk if it is in this context. To get the same thing with just + do it like: (do x := (x + 1) return x) + (do x := (x + x) return x)

4)

a) "When applying reduction rules to terms in some typed variants of the lambda calculus, the ordering in which the reductions are chosen does not make a difference to the eventual result" - Wikipedia b)

succ n = $(\lambda n.\lambda f.\lambda x\ f\ (\ n\ f\ x\ ))\ \underline{n}$

$= \lambda f.\lambda x.\ f\ (\ \underline{n}\ f\ x\ )$

$= \lambda f.\lambda x.\ f\ ((\ \lambda f.\ \lambda x.\ f^{n}(x))\ f\ x\ )$

$= \lambda f.\lambda x.\ f\ ((\ \lambda x.\ f^{n}(x))\ x)$

$= \lambda f.\lambda x.\ f\ (\ f^{n}(x))$

$= \lambda f.\lambda x.\ \ f^{n+1}(x)$

$= \underline{n+1}$

c)

Plus $\underline{m}\ \underline{n} = (\lambda m.\lambda n.\lambda f.\lambda x.\ m\ f\ (\ n\ f\ x\ )\ )\ \underline{m}\ \underline{n}$

$= (\lambda n.\lambda f.\lambda x.\ m\ f\ (\ n\ f\ x\ )\ )\ \underline{n}$

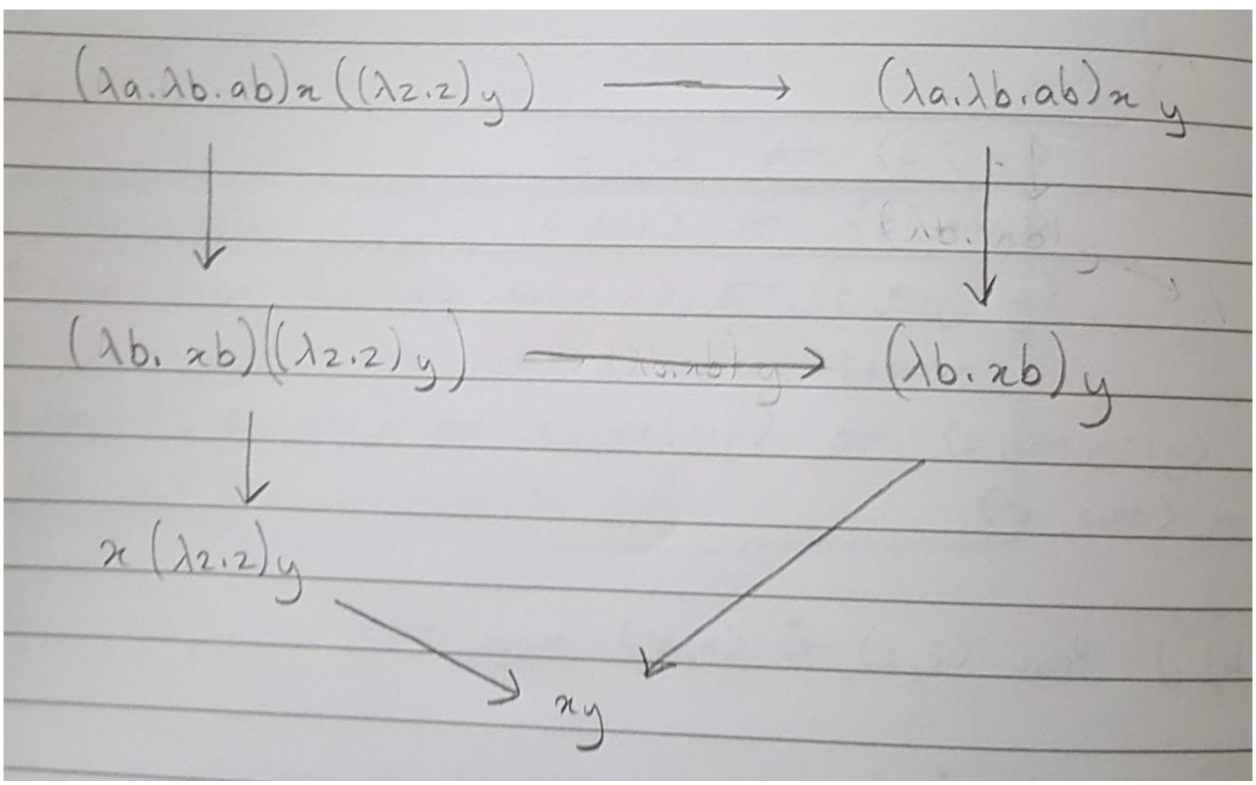
$= \lambda f.\lambda x.\ \underline{m}\ f\ (\ \underline{n}\ f\ x\ )$

$= \lambda f.\lambda x.\ \underline{m}\ f\ (\ (\lambda f.\ \lambda x.\ f^{n}(x))\ f\ x\ )$

$= \lambda f.\lambda x.\ \underline{m}\ f\ (\ (\lambda x.\ f^{n}(x))\ x\ )$

$= \lambda f.\lambda x.\ \underline{m}\ f\ (\ f^{n}(x)\ )$

$= \lambda f.\lambda x.\ (\lambda f.\ \lambda x.\ f^{m}(x))\ f\ (\ f^{n}(x)\ )$



$= \lambda f.\lambda x.\ (\lambda x.\ f^{m}(x))\ (\ f^{n}(x)\ )$

$= \lambda f.\lambda x.\ (f^{m}(f^{n}(x)))$

$= \lambda f.\lambda x.\ (f^{m+n})$

$= \underline{m+n}$

d) APlus $\underline{m}$ $\underline{n}$ $= (\lambda m.\lambda n.\ m\ \mathrm{Succ}\ n)\ \underline{m}\ \underline{n}$

$= \underline{m}\ \mathrm{Succ}\ \underline{n}$

$= (\lambda f.\ \lambda x.\ f^{m}(x))\ \mathrm{Succ}\ \underline{n}$

$= \mathrm{Succ}^{m}\ \underline{n}$

$= \mathrm{Succ}^{m-1}\ \mathrm{Succ}\ \underline{n}$

$= \mathrm{Succ}^{m-1}\ \underline{n+1}$ (By result shown in b)

$= \underline{n+m}$ (By repeated application of Succ)

APlus $\underline{m}$ $\underline{n}$ reaches the same form as Plus $\underline{m}$ $\underline{n}$, thus it can be said that they're beta equivalent.

e)
Mult = λm.λn. m (Plus n) 0

Plus n to $\underline{0}$, m times