Concurrent Object-Oriented Programming

Exercise session 8:
Encoding Objects in the Lambda-Calculus.

1. The paradoxical combinator:
The lambda function:
\[ Y \equiv \lambda f. (\lambda x. f(x,x))(\lambda x. f(x,x)) \]
Is known as: the paradoxical combinator, or the fixpoint combinator.

Reduce the term YF, for a given parameter (called functional) F. What can you observe?
Can you reduce furthermore?
Use it on the following definition of the factorial, for n=2:

\[ \text{Fact} \equiv \lambda f. \lambda n. \text{if (iszero n) 1 (mult n fact(pred n))} \]

\[ \text{factorial} \equiv Y \text{ Fact} \]

2. The encoding of objects:
What do you need to represent an object? What problem do you encounter?
Show how you can construct objects using recursive functions.
Try it on the following class:

Class Cell
Feature
x: int;
get is return X end
set (n) is x := n end;
bump is x := x + 1 end
end Class Cell

Reduce a configuration where two successive calls bump() and get() are made on such an object.