Using call-by-value reduction:

XOR p (NOT p)
= (λpq.p (NOT q) q) p (NOT p) (applying XOR definition)
= (~~λq~~.p (NOT q) p) (NOT p) (applying p for q)
= p (NOT (NOT p)) (applying NOT for p and (NOT p))
= p ((λpqr.p r q) p (NOT p)) (applying NOT definition for (NOT p))
= p ((λqr.p (NOT p) q) p (NOT p)) (applying p for p and (NOT p))
= p (p (NOT p) (NOT p)) (applying (λqr.p (NOT p) q) for p and (NOT p))
= p (p (λpqr.p r q) (NOT p) (NOT p)) (applying (NOT p) for q)
= p (p (λr.(NOT p) r (NOT (NOT p))) (NOT p)) (applying λpqr.p r q for p, (NOT p) and (NOT p))
= p (p (NOT p) (NOT (NOT p))) (applying λr.(NOT p) r (NOT (NOT p)) for r)
= p (NOT p)

Using call-by-name reduction:

XOR p (NOT p)
= (λpq.p (NOT q) q) p (NOT p) (applying XOR definition)
= p (NOT (NOT p)) q (applying p for p and (NOT p))
= p ((λpqr.p r q) p (NOT p)) q (applying NOT definition for (NOT p))
= p ((λqr.p (NOT p) q) p (NOT p)) q (applying p for p and (NOT p))
= p (p (NOT p) (NOT p)) q (applying (λqr.p (NOT p) q) for p and (NOT p))
= p (p (λpqr.p r q) (NOT p) (NOT p)) q (applying (NOT p) for q)
= p (p (λr.(NOT p) r (NOT (NOT p))) (NOT p)) q (applying λpqr.p r q for p, (NOT p) and (NOT p))
= p (NOT p) q (applying λr.(NOT p) r (NOT (NOT p)) for r)
= p (NOT p)

In both call-by-value and call-by-name reductions, the final result is p (NOT p), which represents the logical operator for the boolean equality.