**SOEN 331 Assignment 2**

Michael Bilinsky 26992358

David Bastien 26948553

**2.1 Liskov’s Principle**

Liskov’s substitution principle states that “in a computer program, if S is a subtype of T, then objects of type T may be replaced with objects of type S.” This means that any subclass can be used where the superclass is expected. With regards to contractual specifications, the principle holds that preconditions cannot be strengthened in a subtype, postconditions cannot be weakened in a subtype, and invariants must be preserved in a subtype. In our assignment, no methods are overridden so preconditions and postconditions are never substituted. Some invariants are strengthened since the verifications of the subtypes also protect against possible violations specified in the supertype. For example, the sameHeight() invariant in the PerfectBinaryTree also checks for the isTwoOrNoLeaf() invariant in the supertype FullBinaryTree since a node missing a leaf fails the sameHeight() check. However, the height() != 0 invariant in the BinTree supertype is not replicated in the subtype trees.