**Assignment 8: Building a Family Tree in Prolog**

**Rinku Gopali**

**University of the Cumberlands**

**Advanced Programming Languages (MSCS-632-M50)**

**Dr. Vanessa Cooper**

**04/23/2025**

**Introduction**  
This assignment involved designing a family tree in Prolog to explore logical inference and recursive rule definitions. The goal was to represent family relationships and enable complex queries using Prolog’s pattern matching and rule-based logic.

**Implementation Details**  
Basic facts were defined for parent-child relationships and gender using predicates like parent/2, male/1, and female/1. Derived relationships were implemented using logical rules. For example, sibling/2 identifies individuals with shared parents, while cousin/2 uses sibling relationships between parents. Recursive rules were introduced for ancestor/2 and descendant/2, enabling multigenerational traversal. Additional rules such as father/2, mother/2, son/2, and grandparent/2 added depth to the tree.

**Challenges Faced**  
A key challenge was writing recursive rules for ancestor and descendant relationships without creating infinite loops. Ensuring correctness in conditions such as X \= Y in sibling detection was critical to avoid self-matching. Understanding Prolog’s backtracking and inference engine also required careful structuring of rules to ensure efficient query resolution.

**Conclusion**  
This project demonstrated the power of Prolog in handling hierarchical and relational data using logical inference. By modeling a multi-generational family tree, we effectively leveraged recursion and rule-based definitions to answer complex queries. The assignment enhanced understanding of Prolog’s declarative paradigm and its applicability in AI and knowledge representation.