# CS365 Lab C - Part 1: Design Plan

## 1. Introduction

This design plan outlines the architecture and workflow of a Python program to implement a decision tree learning algorithm. The goal is to build a decision tree that classifies data based on a tab-delimited dataset, as described in Chapter 18 of the course textbook.

## 2. High-Level Design Overview

The system will consist of the following main components:

- Data Loader: Loads tab-delimited files and parses attributes and labels.
- Tree Builder: Implements the recursive decision tree construction algorithm using entropy-based splits.
- Tree Printer: Outputs the decision tree in a human-readable format.
- Cross-Validator: Implements leave-one-out cross-validation for accuracy assessment.

## 3. Components Description

- Data Loader

Reads a dataset from a tab-delimited file. The first line contains the feature names and the last column is the classification target (yes/no). Each data row is parsed into a dictionary for flexibility.

- Tree Builder

Constructs the decision tree recursively using the Decision-Tree-Learning algorithm. It chooses the best attribute based on information gain. Leaf nodes are created when entropy is zero or no attributes remain.

- Tree Printer

Prints the tree using indentation to indicate depth levels. Each node displays the attribute and decision path clearly.

- Cross-Validator

Implements leave-one-out cross-validation. For each instance, builds the tree using the remaining data and tests prediction accuracy. Aggregates overall accuracy.

# CS365 Lab C - Part 1: Design Plan

## 4. Notes for Implementation

- Handle missing or unique attribute values gracefully during tree building.
- Avoid branching on attributes that do not reduce entropy.
- When ties occur in class labels, default to 'no'.
- Use math.log2 for computing entropy.

## 5. Tools and Libraries

Python 3 will be used with standard libraries including:

- csv (for parsing tab-delimited files)
- math (for entropy calculations)
- sys (for command-line arguments)