**CS 300 Project One Revised Milestone Two**

**Huan Ai**

**4/13/2025**

**// Course Structure Definition**

struct Course {

std::string courseCode; // Key field (e.g., "MATH201")

std::string title;

std::vector<std::string> prerequisites;

};

**// Hash Function for Course Codes**

hashFunction(String courseCode) Returns size\_t

Create hash value = 0

For each character in courseCode

hash = (hash \* 31) + ASCII value of character

Return hash % TABLE\_SIZE // Ensures index fits within table bounds

End

**// File Parsing & Hash Table Insertion**

loadCoursesIntoHashTable(String filePath) Returns unordered\_map<string, Course>

Create empty hash table named courseTable

Open input file

If file fails to open

Output "Error: File not found"

Return empty table

While getline(file, currentLine)

Create stringstream from currentLine

Course newCourse

vector<string> tokens

string token

// Tokenize line

While getline(stringstream, token, ',')

token = trimWhitespace(token)

tokens.push\_back(token)

// Validate minimum 2 tokens (code + title)

If tokens.size() < 2

Output "Invalid line: " + currentLine

Continue

// Populate course object

newCourse.courseCode = tokens[0]

newCourse.title = tokens[1]

// Add prerequisites (tokens 2..n)

For i from 2 to tokens.size()-1

newCourse.prerequisites.push\_back(tokens[i])

// Insert into hash table using custom hash

size\_t hashIndex = hashFunction(newCourse.courseCode)

courseTable[hashIndex] = newCourse // Linear probing handled automatically

Return courseTable

End

**// Enhanced Search Function**

searchCourse(CourseHashTable courses, String targetCode) Returns Course\*

// Calculate hash index

size\_t hashIndex = hashFunction(targetCode)

// Direct hash table lookup

If courses.find(hashIndex) exists AND courses[hashIndex].courseCode == targetCode

Return &courses[hashIndex]

// Not found case

Return nullptr

End

**// Prerequisite Validation with Hash Table**

validatePrerequisites(CourseHashTable courses)

For each bucket in courses

For each prerequisite in bucket.course.prerequisites

If searchCourse(courses, prerequisite) returns nullptr

Output "Missing prerequisite: " + prerequisite

Return False

Return True

End

**// Print Function Using Hash Table**

printCourseInfo(CourseHashTable courses, String courseCode)

Course\* foundCourse = searchCourse(courses, courseCode)

If foundCourse is nullptr

Output "Course not found"

Else

Output "Course: " + foundCourse->courseCode + " - " + foundCourse->title

If foundCourse->prerequisites not empty

Output "Prerequisites:"

For each prereq in foundCourse->prerequisites

Output "- " + prereq

End

**// Main Program**

Main()

Initialize CourseHashTable

// Load data

String filePath = getInput("Enter file path [default: courses.csv]: ")

If filePath is empty

filePath = "courses.csv"

courseTable = loadCoursesIntoHashTable(filePath)

// Validate prerequisites

If not validatePrerequisites(courseTable)

Exit with error

// Interactive search

While True

String input = getInput("Enter course code (or 'exit'): ")

If input == "exit"

Break

printCourseInfo(courseTable, input)

End