7-1 Project Two

Duane Wegner

Southern New Hampshire University

CS-300 DSA: Analysis and Design

Prof. Jack Lusby (MS CS)

October 20, 2024

**7-1 Project Two**

**Screenshots:**

**Choice 1:**

**A computer screen shot of a computer program

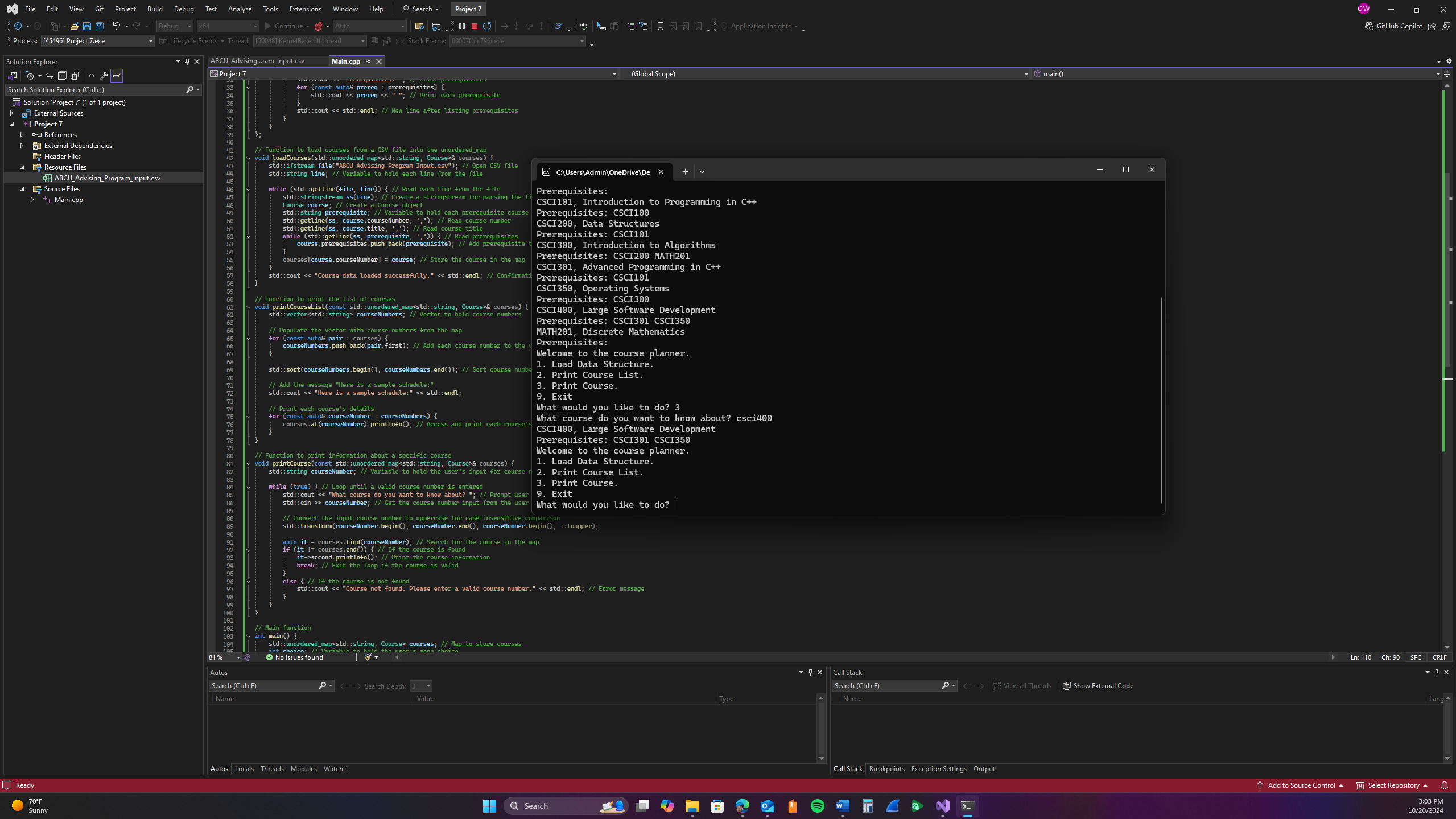
Description automatically generated**

**Choice 2:**

**A screen shot of a computer

Description automatically generated**

**Choice 3:**

****

**Choice 9:**

**A computer screen with a black screen

Description automatically generated**

**Code:**/\*\*

\* This program is a course planner that loads course data from a CSV file,

\* allowing users to print a list of courses and view course details.

\* Author: Duane Wegner

\* Date: October 14, 2024

\*/

#include <iostream>

#include <unordered\_map>

#include <vector>

#include <string>

#include <algorithm>

#include <fstream>

#include <sstream>

#include <cctype>

class Course {

public:

std::string courseNumber; // Course number (e.g., "CS101")

std::string title; // Title of the course (e.g., "Introduction to Programming")

std::vector<std::string> prerequisites; // List of prerequisite courses

// Function to print course information

void printInfo() const {

std::cout << courseNumber << ", " << title << std::endl; // Print course number and title

if (prerequisites.empty()) {

std::cout << "Prerequisites: None" << std::endl; // Indicate no prerequisites

}

else {

std::cout << "Prerequisites: "; // Print prerequisites

for (const auto& prereq : prerequisites) {

std::cout << prereq << " "; // Print each prerequisite

}

std::cout << std::endl; // New line after listing prerequisites

}

}

};

// Function to load courses from a CSV file into the unordered\_map

void loadCourses(std::unordered\_map<std::string, Course>& courses) {

std::ifstream file("ABCU\_Advising\_Program\_Input.csv"); // Open CSV file

std::string line; // Variable to hold each line from the file

while (std::getline(file, line)) { // Read each line from the file

std::stringstream ss(line); // Create a stringstream for parsing the line

Course course; // Create a Course object

std::string prerequisite; // Variable to hold each prerequisite course

std::getline(ss, course.courseNumber, ','); // Read course number

std::getline(ss, course.title, ','); // Read course title

while (std::getline(ss, prerequisite, ',')) { // Read prerequisites

course.prerequisites.push\_back(prerequisite); // Add prerequisite to the course

}

courses[course.courseNumber] = course; // Store the course in the map

}

std::cout << "Course data loaded successfully." << std::endl; // Confirmation message

}

// Function to print the list of courses

void printCourseList(const std::unordered\_map<std::string, Course>& courses) {

std::vector<std::string> courseNumbers; // Vector to hold course numbers

// Populate the vector with course numbers from the map

for (const auto& pair : courses) {

courseNumbers.push\_back(pair.first); // Add each course number to the vector

}

std::sort(courseNumbers.begin(), courseNumbers.end()); // Sort course numbers alphanumerically

// Add the message "Here is a sample schedule:"

std::cout << "Here is a sample schedule:" << std::endl;

// Print each course's details

for (const auto& courseNumber : courseNumbers) {

courses.at(courseNumber).printInfo(); // Access and print each course's details

}

}

// Function to print information about a specific course

void printCourse(const std::unordered\_map<std::string, Course>& courses) {

std::string courseNumber; // Variable to hold the user's input for course number

while (true) { // Loop until a valid course number is entered

std::cout << "What course do you want to know about? "; // Prompt user for a course number

std::cin >> courseNumber; // Get the course number input from the user

// Convert the input course number to uppercase for case-insensitive comparison

std::transform(courseNumber.begin(), courseNumber.end(), courseNumber.begin(), ::toupper);

auto it = courses.find(courseNumber); // Search for the course in the map

if (it != courses.end()) { // If the course is found

it->second.printInfo(); // Print the course information

break; // Exit the loop if the course is valid

}

else { // If the course is not found

std::cout << "Course not found. Please enter a valid course number." << std::endl; // Error message

}

}

}

// Main function

int main() {

std::unordered\_map<std::string, Course> courses; // Map to store courses

int choice; // Variable to hold the user's menu choice

do {

std::cout << "Welcome to the course planner." << std::endl; // Welcome message

std::cout << "1. Load Data Structure." << std::endl; // Option to load courses

std::cout << "2. Print Course List." << std::endl; // Option to print course list

std::cout << "3. Print Course." << std::endl; // Option to print specific course

std::cout << "9. Exit" << std::endl; // Option to exit the program

std::cout << "What would you like to do? "; // Prompt for user choice

std::cin >> choice; // Get user choice

switch (choice) {

case 1:

loadCourses(courses); // Load course data

break;

case 2:

printCourseList(courses); // Print list of courses

break;

case 3:

printCourse(courses); // Print specific course details

break;

case 9:

std::cout << "Thank you for using the course planner!" << std::endl; // Exit message

break;

default:

std::cout << choice << " is not a valid option." << std::endl; // Invalid option message

}

} while (choice != 9); // Continue until the user chooses to exit

return 0; // Return statement for main

}