Assignment 2 - Planning a Programming Solution

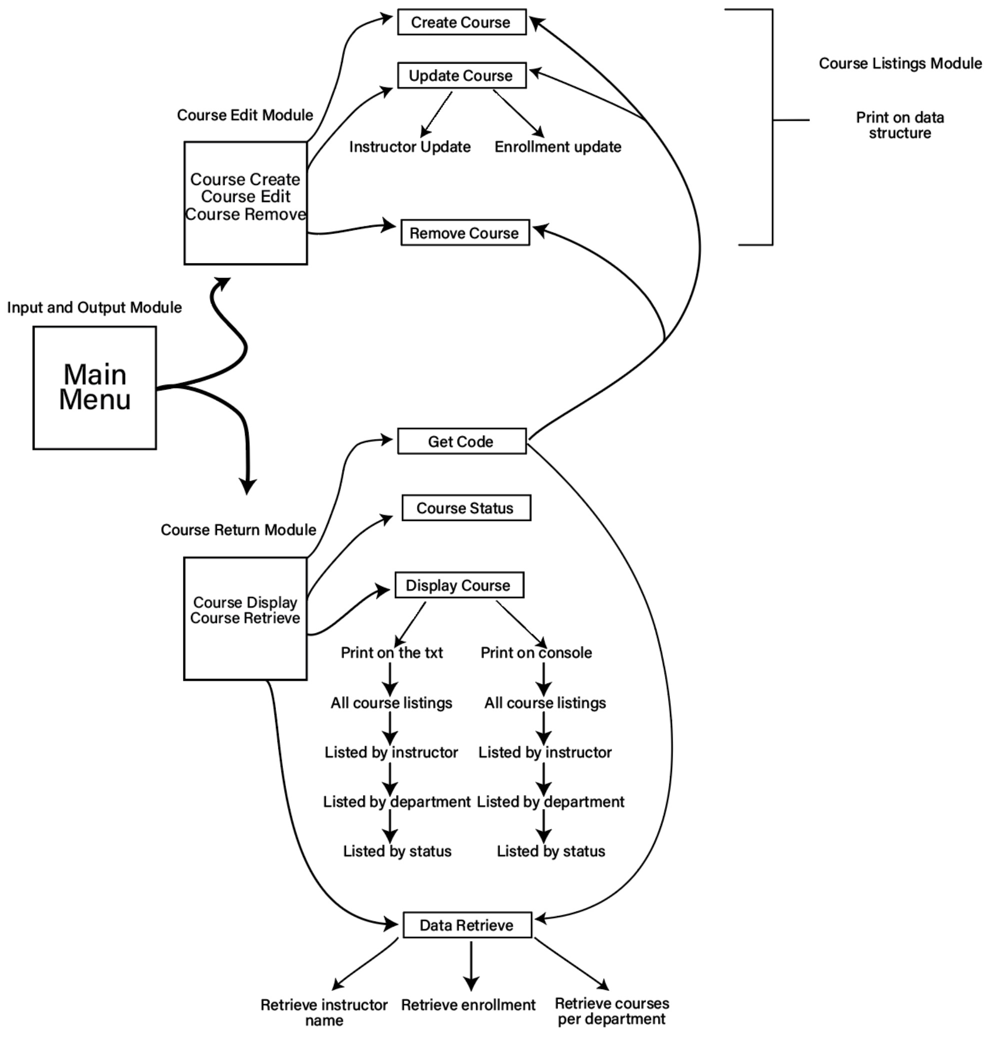
CISC 121 - Amy Brons

This document accompanies the planning presentation submitted in this assignment. It it written to follow the word cloud step-wise refinement document.

Reasoning, Planning and Module Breakdown

To create this program I looked at the ways in which the data needs to interact with each other. For start I set one module a side for input and output. After determining the different options that the user needs to be able to select, I determined that there is two types of choices; editing and viewing. In editing, the user needs to be able to directly affect the data. In retrieval and display, the user only needs to be able to view the data, or copy select info into a new text file. This means that I determined there to be two more modules. Finally, the last module came from needing a central place to have a new data structure that can be interacted from the other modules.

To help plan and visualize my program, I created a graph to help plot data flow. Here is a computerized version of my sketches:



Module 1 - InputOutput

userInput()

Description : This function directs users through the selection menus to get a numerical selection of what task the user wants to accomplish on the system.

Input: Prompts users to input 1 or 2 times to get the appropriate selection of task for the user. The input is an integer between 1 and 5.

Output: The program outputs the user selection for use in the programOutput() function.

programOutput()

Description: This function decides which module and function to run, based on user input in the previous function.

Input: The function runs the userInput() function to find the appropriate input. The input wil be an integer between 1 and 11. The number then is put through the function and the module and function are decided from there.

Output: The selected module and function is run, and their individual outputs act as this functions out put.

main()

Description: Runs programOutput()

Module 2 - courseListings

dataStruct()

Description: This function takes the txt file of course1.txt and puts the information into an editable list of lists.

Input: The input will be the read courses.txt file, which is input within the function.

Output: An editable data structure in the form of a list of list.

Module 3 - courseEdit

createCourse()

Description: Creating a new list to add to our list of lists data structure.

Input: Users are prompted to answer a series of questions to gather information for the list.

Output: The inputted data, in the form of a list. Output is appended to the data structure.

updateCourse(prompt)

Description: Based on the prompt input form the InputOutput module, the function decides to either allow the user to update the number of students enrolled in a course, or the instructor of the course.

Input: The function takes the prompt that the user Inputted and prompts them for more input in the form of the class code. After that, the function also either asks for the updated enrolment number or the updated instructor.

Output: The function outputs the updated class info, and prints the update for user readability.

removeCourse()

Description: Based on user input, one course is removed from the data structure.

Input: Class code is inputted by the user when prompted.

Output: The class is removed from our data structure and a confirmation is printed to the user.

Module 4 - courseReturn

getCode()

Description: Checks to confirm that the user inputted code is the correct format.

Input: Prompts users to input code.

Output: Returns code for use.

isOpen(classSize, enrolled)

Description: Checks to see if there is open spots in the course to enrol more students.

Input: Parameters of the class size and the enrolled students number are taken in.

Output: If there is room then the string “Available” is returned, if full the the string “Full” is returned.

courseDisplay(prompt)

Description: Displays the course information in the txt file in one of four ways; as all courses, by instructor, by department or by availability.

Input: Prompts users to ask if they want to write within a text file, or display on the console. Takes the prompt input from the InputOutput module, and takes information from the txt file.

Output: Prints a list of courses, three different ways.

numberRetrive(prompt)

Description: Displays data in three different ways; the instructor name for a course, the number of students enrolled in a course, or the courses within a certain department.

Input: Input a course code or a department name. The other input is the txt file.

Output: Prints the course information found in the file.