Project 3

Focus:

The purpose of this project is to comprehensively bring together all of the things that you have learned so far from Modules 5 and 6. The project expands on what you created in the second project by modifying it to use functions and arrays.

Instructions:

For this project you are going to be working with information that you are familiar with, a class schedule, billing statement, and of course the concepts to writing a program that you have learned this far. For this project you are going to be working on modifying the code that you wrote in Project 2.

For project 3, you will need to break down your project 2 into functions and make additional modifications as specified in the instructions.

The following information should be defined in your program:

1. The tuition per credit hour is $263.00.
2. The overall fees for a student's registration is $162.50
3. Two different scholarship amounts, Academic Scholarship should be set to $500, while the Foundation scholarship should be set to $750
4. You will be working with four arrays; each array is graphically shown below with the information that should be stored in it. Three are string arrays and one is an integer array. Each cell in the row shows the element that should be found in the array. The name that you should use for your array is shown on the first column. **Remember, that the count of an array begins with 0 not 1.** If nothing is shown in the cell, then you should initialize it to be "blank". Do not chance the way that the items are displayed, if you do it will impact the rest of your program.

courseNo	310	452	305	308
mathClass	Matrices and Linear Algebra	Introductory Real Analysis	Calculus III	Discrete Math
infoTechClass	Data Analysis and Visualization	Senior Capstone Project	Networking Fundamentals	Network Architecture
compSciClass	Ethics	Senior Capstone Project	Concepts of Programming	Computer Science II

5. You will have a 5th string array where you will store the classes the user has selected. One element in this array will consist of course prefix, number, and title.
6. Classes will be 3 credit hours, 2 credit hours, or 1 credit hour. CS 310 Ethics is a 1 credit hour class. ITE 305 Networking Fundamentals and ITE 308 Network Architecture are 2 credit hours, and all other courses are 3 credit hours.
   **Note: Your program, will need to use the different pieces of information, course prefix, course number to determine the number of credit hours to "assign" for the schedule and billing portion of the program. **

Your program should start by asking the user to enter the following:

1. Their first name
2. Their last name
3. Their student ID number. Only the numbers after any leading 0's.

Once the user has entered their information you will then use a switch statement to provide the user with a Menu where the user can decide what he or she wants to do next. The menu and each of its options will be broken into functions. You will need to have the following functions:

1. Menu – this function will display the menu to the user and allow for the user to enter his/her option. You must validate that the user has selected a valid menu option before continuing.

You will then have a function for each of the menu options.

2. **SelectClasses** – this function will have multiple steps associated to it.
   a. First the user should be asked how many classes he/she wants to select. There is a total of 12 classes the user can choose from (consider this when creating your 5th array).
   b. Once the user has entered how many classes they want to enter you must then give the user the option to select the course prefix CS, ITE, or MA. The course prefix will allow for you to determine which class array to work with.
   c. When the user has specified the course prefix, you will then ask the user for the course number. You must verify that the number exists in the courseNo array. Once the element in the courseNo array is found, the "index" will specify the index for the class in the class array. You will then access that specific element to obtain the name of the course.
   d. When you have each of these three steps you will be able to populate your 5th array with the classes that have been selected.

      **Do not allow the user to select a class more than once. **

3. **Scholarship** – this function will allow the user to select the type of scholarship if they are receiving one. If not, then the scholarship value should be set to 0. Your program should then determine the amount that will be used when the bill is calculated.
   Note: The user is not setting the scholarship amount. Your program does this. The user only select which of the two scholarships he/she is receiving, if any.
4. **ViewSchedule** – this function will display the student schedule on the screen. The student schedule should be similar to that from project 1. You will use information that was gathered through the class selection function to then properly display the schedule with the appropriate tabular formatting. You must also include the credit hours for the classes and the overall credit hour that the student is enrolled in.
5. **SaveSchedule** – this function will write the student schedule to a file called Schedule.txt. The student schedule should be similar to that from project 1. You will use information that was gathered through the class selection function to then properly display the schedule with the appropriate tabular formatting. You must also include the credit hours for the classes and the overall credit hour that the student is enrolled in.
   This function will also be called by PrintDetails. The purpose of functions is to be able to reuse them. Think about how you will specify the file that the information is written to.
6. **ViewBill** – this function will display the bill based on the schedule that the student has put together. The display of the bill should be similar to that of project 1. You will use the information that was gathered to determine the cost of tuition, the fees, the scholarship, and the overall cost. You need to format this information appropriately since you are working with money in this function, and be sure that your calculations are correct.
7. **SaveBill** - this function will write the bill to a file called Bills.txt, based on the schedule that the student has put together. The file should be similar to the formatting used in project 1. You will use the information that was gathered to determine the cost of tuition, the fees, the scholarship, and the overall cost. You need to format this information appropriately since you are working with money in this function, and be sure that your calculations are correct.
   This function will also be called by PrintDetails. The purpose of functions is to be able to reuse them. Think about how you will specify the file that the information is written to.
8. **PrintDetails** – this function will write all the details to a file called Details.txt. This file will include the welcome statement, the schedule and bill. There will not be the option to view all of the information on the screen, as in project 2. Be sure that you use the appropriate output formatting. This function will need to call the function SaveSchedule and SaveBill. You should NOT be re-writing this code. The purpose of functions is to be able to reuse them. Think about how you will specify the file that the information is written to.
9. **Exit** – this function should provide a thank you message and end the program.

In the first project you were asked to display all of the information to the user in one print out. This is the option that the user will have with the View Details menu option, however, the user will be able to go through each individual piece and only obtain the information that they require if they are not interested in seeing the whole thing. Keep in mind, that the order in which the

user chooses to perform specific tasks associated to the menu are important. For example, if the user has not selected classes then the program should not allow for the user to View a Schedule. When it comes to a Bill, if the user has not selected classes but has entered scholarship information, the bill should be displayed as a "credit" for the scholarship. If the user has not entered a scholarship and has not selected classes, then the user should not be able to view the bill. If the user selects View Details, but does not have a schedule or a bill, then the user should still be provided a copy of their acceptance.

The display of the information should follow the format that you completed in Project 1 and 2. Both the schedule and the bill should be found in a tabular format using output formatting, and the letter should populate from information that the user has entered.

Concepts to keep in mind:

1. Assign values to all **constant** variables that are associated with information that will NOT change.
2. Be sure that you are working with the decision making process correctly (if, if/else, nested statements).
3. Only use switch statements for your menus.
4. Be sure that you select appropriate loops where needed. Think about the type of look that you need to use beforehand (counted, pre or posttest)
5. You must have validation throughout the program. Be sure that if the user enters information incorrectly that he/she has the opportunity to fix that entry before the program continues.
6. Remember you are dealing with money therefore should only have 2 decimal places.
7. Every menu option that the user selects should allow for them to go back to the main menu and do something else before they choose to exit the program.
8. Be sure that with your functions you are using both pass by value and pass by reference. You **should NOT** declare all variables globally.
9. There will be points associated to the creativity of your display. You can follow the format that I have provided on the sample output to be sure that you implement the output formatting that is required. You may use escape characters, but points are associated to the output formatting commands and not the escape characters.
10. **Do not use any other concept that we have not covered!**

Deliverables

1. Flowchart
2. Source file
3. PDF with screen shots
4. Schedule.txt
5. Bill.txt
6. Details.txt