COP 2334 Programming Project 2

# Project Outcomes

Develop a C++ program that

* displays messages to the monitor and reads data from the keyboard using cout and cin and the insertion and extraction operators,
* creates and uses different types of variables,
* performs assignment operations,
* performs arithmetic operations,
* performs multiway branches using if-else and switch statements, and
* performs iteration using while, do-while, and/or for loops.

# Program Description

Write a program that

* displays a menu of options to allow the user to select a prescription drug payment plan,
* prompts the user to enter an estimated number of prescriptions that will be filled in one year,
* displays the estimated annual cost of the prescription drug payment plan, based on plan selected estimated number of prescriptions
* allows the user to generate as many estimates as desired.

# Plan Data

The plan monthly premiums, deductible, and copay are given in the table below

| Plan | Monthly Premium | Deductible | Copay |
| --- | --- | --- | --- |
| A | $10.00 | $200.00 | 40% |
| B | $25.00 | $100.00 | 20% |
| C | $50.00 | $0.00 | 0% |

Each prescription costs $50.00.

# Background

For the purpose of this assignment, the following definitions will be used.

* **Monthly premium**: a fixed cost that must be paid whether or not any prescriptions are filled. Multiply the monthly premium by 12 to get the annual premium.
* **Deductible**: the minimum amount of prescription costs that that must be paid before the plan will begin paying a share of the costs.
* **Copay**: the share of the costs above the deductible that is paid by the user.

Using the data for Plan A in the table above, and 20 prescriptions, the cost would be calculated as follows:

annual\_premium = monthly\_premium x 12

= $10.00 X 12

= $120.00

prescription\_cost = 20 x cost\_per\_prescription

= 20 x $50.00 = $1000.00

over\_deductible = prescription\_cost - deductible

= $1000.00 - $200.00

= $800.00

cost = annual\_premium + deductible +

over\_deductible x copay

= $120.00 + $200.00 + $800.00 x 0.4

= $640.00

# Program Requirements

1. The source code file must begin with header comments that include file name, author, email, project number, description, and date modified.
2. When the program starts, display a menu of payment plan options. The format of the plan options is at your discretion, but must contain the plan letter (A, B, or C), monthly premium, deductible, and copay percentage. A possible format might be:

Plan A

Monthly premium $10.00

Deductible $200.00

Copay 40%

Plan B

Monthly premium $25.00

Deductible $200.00

Copay 40%

Plan C

Monthly premium $50.00

Deductible $0.00

Copay 0%

1. Prompt the user to select a plan by entering the character associated with the plan. For example, to select Plan C, the user would enter the 'C' or 'c' character. The prompt must be displayed ***exactly*** as follows, with the user input shown in ***bold italic***:

Please select a plan (enter A, B, or C): ***C***

* 1. Note that there must be a single space after the colon and the user input must appear on the same line as the prompt.
  2. The program must recognize both uppercase and lowercase letters for the plan option. For example, user input of both uppercase ‘C’ and lowercase ‘c’ must be recognized as option C.
  3. If the user enters an invalid character (any character other than ‘A', ‘a’, ‘B', ‘b’, ‘C’, or ‘c’), display an appropriate error message, then redisplay the prompt as shown in paragraph 3. Above. Continue to display the error message and prompt until the user enters a valid plan option.
  4. The error message must include the user input in the error message. For example, if the user entered ‘D’, the error message might be:

Sorry, ‘D’ is not a valid plan option. Please try again.

1. Prompt the user to enter the estimated annual number of prescriptions to be filled. The maximum number of prescriptions is 96. The prompt must be displayed ***exactly*** as follows, with the user input shown in ***bold italic***:

Please enter the estimated number of prescriptions (0 to 96): ***12***

* 1. Note that there must be a single space after the colon and the user input must appear on the same line as the prompt.
  2. If the user enters an invalid number (less than 0 or more than 96), display an appropriate error message, then redisplay the prompt as shown in paragraph 4. Above. Continue to display the error message and prompt until the user enters a valid number of prescriptions.
  3. The error message must include the user input in the error message. For example, if the user entered 100, the error message might be:

Sorry, ‘100’ is not a valid number of prescriptions. Please try again.

1. Display the cost for the selected plan and number of prescriptions. The cost for the selected plan must be displayed ***exactly*** as follows:

Plan C cost for 12 prescriptions is $600.00

1. Prompt to ask whether the user would like another plan cost estimate. The prompt must be displayed ***exactly*** as follows, with the user input shown in ***bold italic***:

Would you like another plan cost estimate? (y/n): ***y***

* 1. Note that there must be a single space after the colon and the user input must appear on the same line as the prompt.
  2. Both lowercase and uppercase ‘y’ (either ‘y’ or ‘Y’) must allow another plan cost estimate. Any other character must terminate the program.

1. You must use both a switch and if-else constructs in the program.
   1. The switch construct is useful for a multiway branch with many options, such as when comparing the user’s plan input char to the valid plan options.
2. All numeric values for plan monthly premiums, deductible, and copay must be declared as constants.
3. See the output of the sample runs below for more examples.

# Suggested Plan of Development

1. Create a loop that displays the plan options and displays the prompt that asks whether the user would like another plan cost estimate. This will be the outermost loop. Recommend a do-while loop since the options and prompt for another estimate will need to run at least once. Thoroughly test this loop to ensure that a ‘Y’ or ‘y’ will redisplay the plan options and the prompt and any other character will terminate the program.
2. Create another loop that will prompt for the plan option. This loop should go after the plan options are displayed and before the prompt for another estimate. This loop should continue until the user enters a valid plan option. Again, a do-while loop is recommended since this prompt must be displayed at least once.
3. Create another loop, after the plan option loop that will prompt for estimated number of prescriptions. This loop should be similar to the plan option loop: it will continue until the user enters a valid number of prescriptions.
4. Calculate and display the cost for the selected plan.
5. The structure of your program might look something like this:

// main function

// outermost loop

// display plan options

// plan option loop

// display prompt for plan option

// display error message if invalid plan option

// end plan option loop when valid plan option input

// prescription number loop

// display prompt for plan option

// display error message if invalid prescription number

// end prescription number loop when valid prescription

number input

// calculate selected plan costs

// display selected plan cost

// prompt for another estimate

// end outermost loop when other than ‘y’ or ‘Y’ input

// end main function

# Submission

1. Compile and run your program one last time before submitting it.
2. Review the rubric associated with the project to ensure all functionality is completed.
3. Submit your source code file as described in the assignment in eLearning/Canvas. Your source code file may have any name, as long as it has the .cpp file extension.
4. After you have submitted your source code file, download it to a folder different than your original source code. Compile and run the downloaded source code to ensure that you have submitted the correct source code file.

# Sample Output

User input is shown in ***bold italic***.

**Sample Run 1 – All valid input**

Plan A

Monthly premium $10.00

Deductible $200.00

Copay 40%

Plan B

Monthly premium $25.00

Deductible $100.00

Copay 20%

Plan C

Monthly premium $50.00

Deductible $0.00

Copay 0%

Please select a plan (enter A, B, or C): A

Please enter the estimated number of prescriptions (0 to 96): 20

Plan A cost for 12 prescriptions is $640.00

Would you like another plan cost estimate? (y/n): y

Plan A

Monthly premium $10.00

Deductible $200.00

Copay 40.00%

Plan B

Monthly premium $25.00

Deductible $100.00

Copay 20.00%

Plan C

Monthly premium $50.00

Deductible $0.00

Copay 0.00%

Please select a plan (enter A, B, or C): c

Please enter the estimated number of prescriptions (0 to 96): 10

Plan C cost for 12 prescriptions is $600.00

Would you like another plan cost estimate? (y/n): n

**Sample Run 2 – Invalid plan input**

Plan A

Monthly premium $10.00

Deductible $200.00

Copay 40%

Plan B

Monthly premium $25.00

Deductible $100.00

Copay 20%

Plan C

Monthly premium $50.00

Deductible $0.00

Copay 0%

Please select a plan (enter A, B, or C): D

Sorry, 'D' is not a valid plan option. Please try again.

Please select a plan (enter A, B, or C): B

Please enter the estimated number of prescriptions (0 to 96): 17

Plan B cost for 12 prescriptions is $550.00

Would you like another plan cost estimate? (y/n): N

**Sample Run 3 – Invalid number of prescriptions input**

Plan A

Monthly premium $10.00

Deductible $200.00

Copay 40%

Plan B

Monthly premium $25.00

Deductible $100.00

Copay 20%

Plan C

Monthly premium $50.00

Deductible $0.00

Copay 0%

Please select a plan (enter A, B, or C): a

Please enter the estimated number of prescriptions (0 to 96): -1

Sorry, '-1' is not a valid number of prescriptions. Please try again.

Please enter the estimated number of prescriptions (0 to 96): 97

Sorry, '97' is not a valid number of prescriptions. Please try again.

Please enter the estimated number of prescriptions (0 to 96): 12

Plan A cost for 12 prescriptions is $480.00

Would you like another plan cost estimate? (y/n): z

# Important Notes

1. Projects will be graded on whether they correctly solve the problem and whether they adhere to good programming practices.
2. Projects must be submitted by the time specified on the due date. Projects submitted after that time will get a grade of zero.
3. Please review UWF's academic conduct policy that was described in the syllabus. Note that viewing another student’s solution, whether in whole or in part, is considered academic dishonesty. Further, allowing another student to view your code is considered academic dishonesty. Finally, submitting code obtained through the Internet or other sources, whether in whole or in part, is considered academic dishonesty. All programs submitted will be reviewed for evidence of academic dishonesty, and all violations will be handled accordingly.