QUESTION 5 – FUNCTIONAL PROGRAMMING

MUHAMMED SUWANEH 152120181098

ESKIŞEHIR OSMANGAZI UNIVERSITY

FACULTY OF ENGINEERING AND ARCHITECTURE

DEPARTMENT OF COMPUTER ENGINEERING

COMPUTER PROGRAMMING

(C++)

HOMEWORK REPORT

LECTURER DR. YILDIRAY ANAGÜN

MARCH 2020

CHAPTER 1

SUMMARY OF PROJECT

The given project is about functional programming i.e. how to structure our program when they become by breaking them into smaller functions which handle specific task and becomes easier to follow and comprehend.

In this write-up, I will break down the functions and explain how each of them work.

The given problem is as follows: Your program will be a "quiz" for the user. That is, your program will present the user with a quiz
of arithmetic problems. Each "**play**" of the quiz will be **10 questions**. The user will initially be presented with a short
menu of options on difficulty level.

CHAPTER 2

METHOD OR MODEL USED

As stated earlier, the project is more of a functional based programming and according to the question the program should be able to ask the user 10 questions produced randomly based on the difficulty level. The user is given a second once the first attempts turns out to be wrong.

I designed some functions in additions to the already given functions in the question.

The task of each function is stated below:

`char display();` → This function takes no parameters and displays the difficulty levels i.e. **easy**, **moderate** and **advance** to the user. Based on the level selected by the user, the function returns a character.

`int randomInt(int, int);` → This function is more logic based and takes two parameters(int) which represent max and min based on what level the user chooses. If the user chooses easy for instance the randomIn(max, min) function gets single digits numbers (max), double digits and three digits for moderate and advance levels respectively.

`char generate_Operator();` → This function takes no parameters and generates and returns operators based ob their ASCII i.e. '+' and '-'.

`int perform_Calculation(char, int, int);` → This function performs calculations between the two random operands generated. It takes three parameters. Char which represents the operator generated and two integers(int) which represent the two operands. It returns the answer generated by the operands.

`int display_Problem(char oper, int num1, int num2, int correct_answer);` → This function does the actual job. It is the main part of the program. It takes four parameters. Char represents the operator, integers represent the operands and the correct answer respectively.It also responsible for displaying to user whether an answer is correct or not based on the `bool isCorrect(int users_answer, int correct_answer)` function.It returns users answer.

`bool isCorrect(int users_answer, int correct_answer);` → This function here takes two parameters and they the integers which represent the users answer and the correct answer returned by

`int perform_Calculation(char, int, int);` function. It returns a boolean which is true or false.

`void display_Results(int correct, int wrong)` → This is the final question of the project and it takes two parameters it is responsible for displaying the number of correct and wrong answers given by the user.

CHAPTER 3

SOURCE CODE

```
/* Question 5 */
/* Functional Programming */
#include <iostream>
#include <cstdlib>
#include <ctime>
#define size 10

using namespace std;

char display();
int randomInt(int, int);
char generate_Operator();
int perform_Calculation(char, int, int);
int display_Problem(char, int, int, int);
bool isCorrect(int, int);
void display_Results(int, int);

int main() {

    srand(time(NULL));

    int num1, num2, max, min, correct_answer, get_users_answer;
    bool get_evaluation;
    char level, oper;
    int correct = 0, wrong = 0, n = 0;
    char leave_quiz = ' ';

    cout << "---------------------- QUIZ ----------------------" << "\n\n";

    do {

        level = display();

        cout << endl;

        if (level == 'a' || level == 'A') {

            max = 9; min = 0;
        }
        else if (level == 'b' || level == 'B') {

            max = 999; min = 0;
        }

        else if (level == 'c' || level == 'C') {

            max = 9999; min = 0;
        }

        cout << "---------------------- QUESTIONS ----------------------" << "\n\n";

        do {

            num1 = randomInt(max, min);
            num2 = randomInt(max, min);

            oper = generate_Operator();
            correct_answer = perform_Calculation(oper, num1, num2);
            get_users_answer = display_Problem(oper, num1, num2, correct_answer);

            get_evaluation = isCorrect(get_users_answer, correct_answer);

            if (get_evaluation == true) {

                correct++;
            }

            else {
```

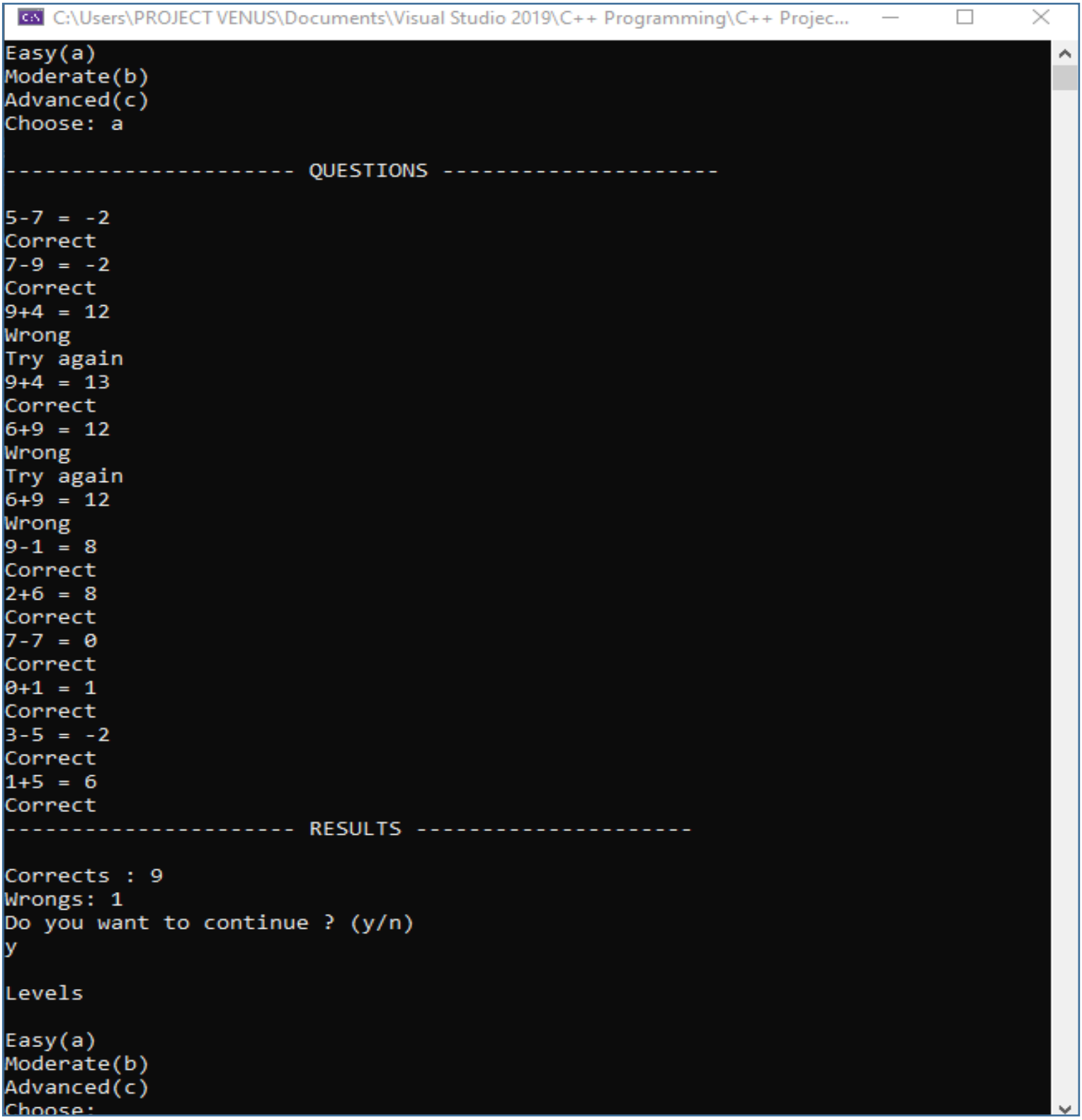
```
                wrong++;
            }

            n++;

        } while (n != 10);

        cout << "---------------------- RESULTS ----------------------" << "\n\n";
        display_Results(correct, wrong);

        cout << "Do you want to continue ? (y/n)" << endl;
        cin >> leave_quiz;

        if (leave_quiz == 'n' || leave_quiz == 'N') {

            cout << "Successfully terminated" << endl;
            cout << "Thanks" << endl;
            break;
        }

    } while (leave_quiz != 'n');

    cout << endl;
    system("pause");
}
```

```
char display() {

    char choice;

    do {
        cout << endl << "Levels" << "\n\n";
        cout << "Easy(a)" << endl;
        cout << "Moderate(b)" << endl;
        cout << "Advanced(c)" << endl;
        cout << "Choose: ";
        cin >> choice;

        if (choice == 'a' || choice == 'A' || choice == 'b' || choice == 'B' || choice == 'c' || choice == 'C') {

            break;
        }

    } while (1);

    return choice;
}

int randomInt(int max, int min) {

    int num(-1);

    while (num < min || num > max)
        num = rand();

    return num;
}

```

```
char generate_Operator() {

    char oper = 1;

    while (oper != 43 && oper != 45)
        oper = rand();

    return oper;
}


int perform_Calculation(char oper, int num1, int num2) {

    switch(oper)
    {

    case 43:
            return num1 + num2;
        break;
    case 45:
            return num1 - num2;
        break;
    default:
        cout << "Syntax error" << endl;
        return 0;
        break;
    }
}
```

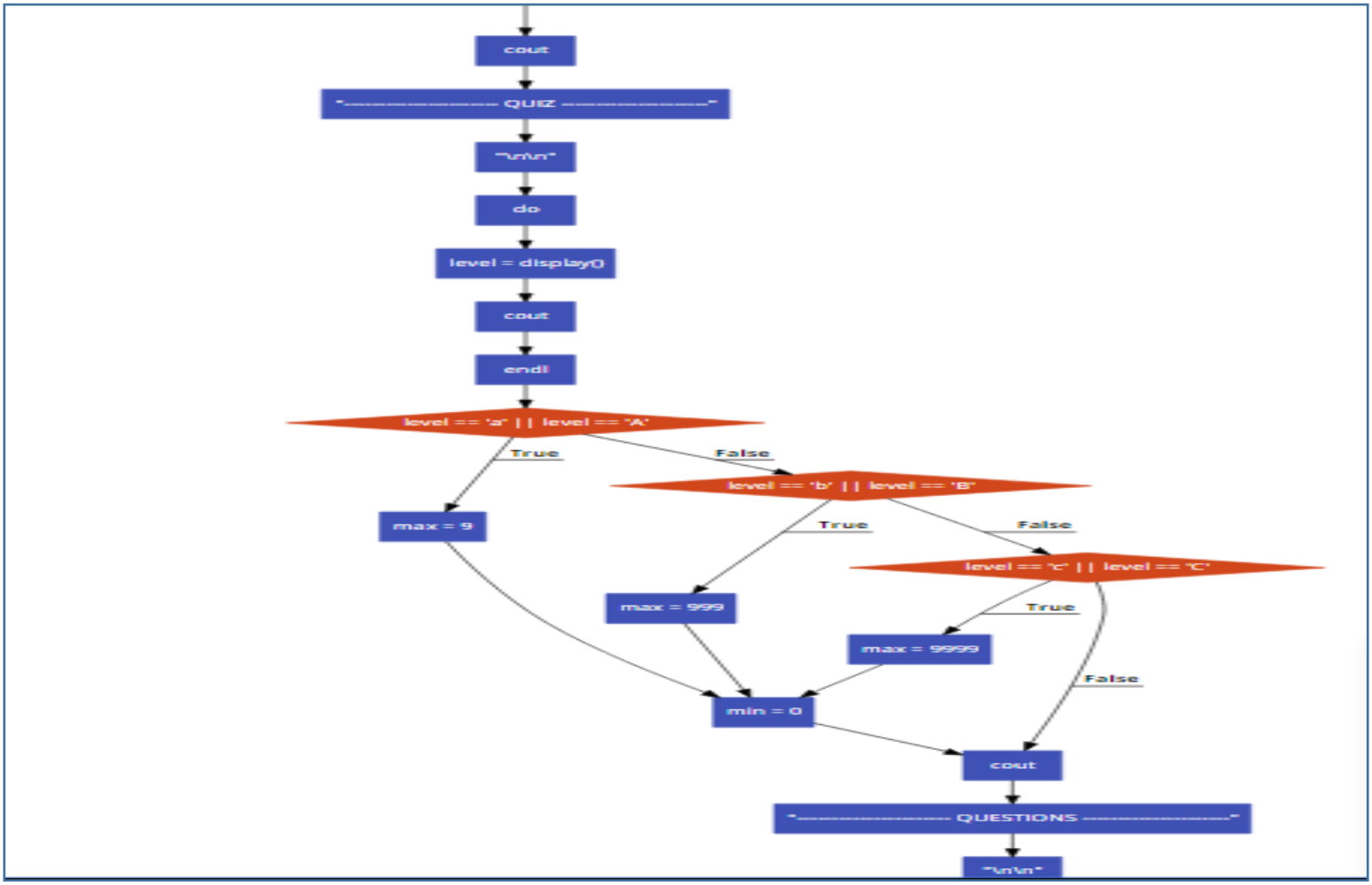
```cpp
int display_Problem(char oper, int  num1, int num2, int correct_answer) {

    int answer, i;

    for (i = 1; i <= 2; i++) {
        if (i == 2) {
            cout << "Try again" << endl;
        }
        cout << num1 << oper << num2 << " = ";
        cin >> answer;
        cin.ignore();

        if (correct_answer == answer) {
            cout << "Correct" << endl;
            break;
        }

        else {

            cout << "Wrong" << endl;
        }
    }

    return answer;
}
```

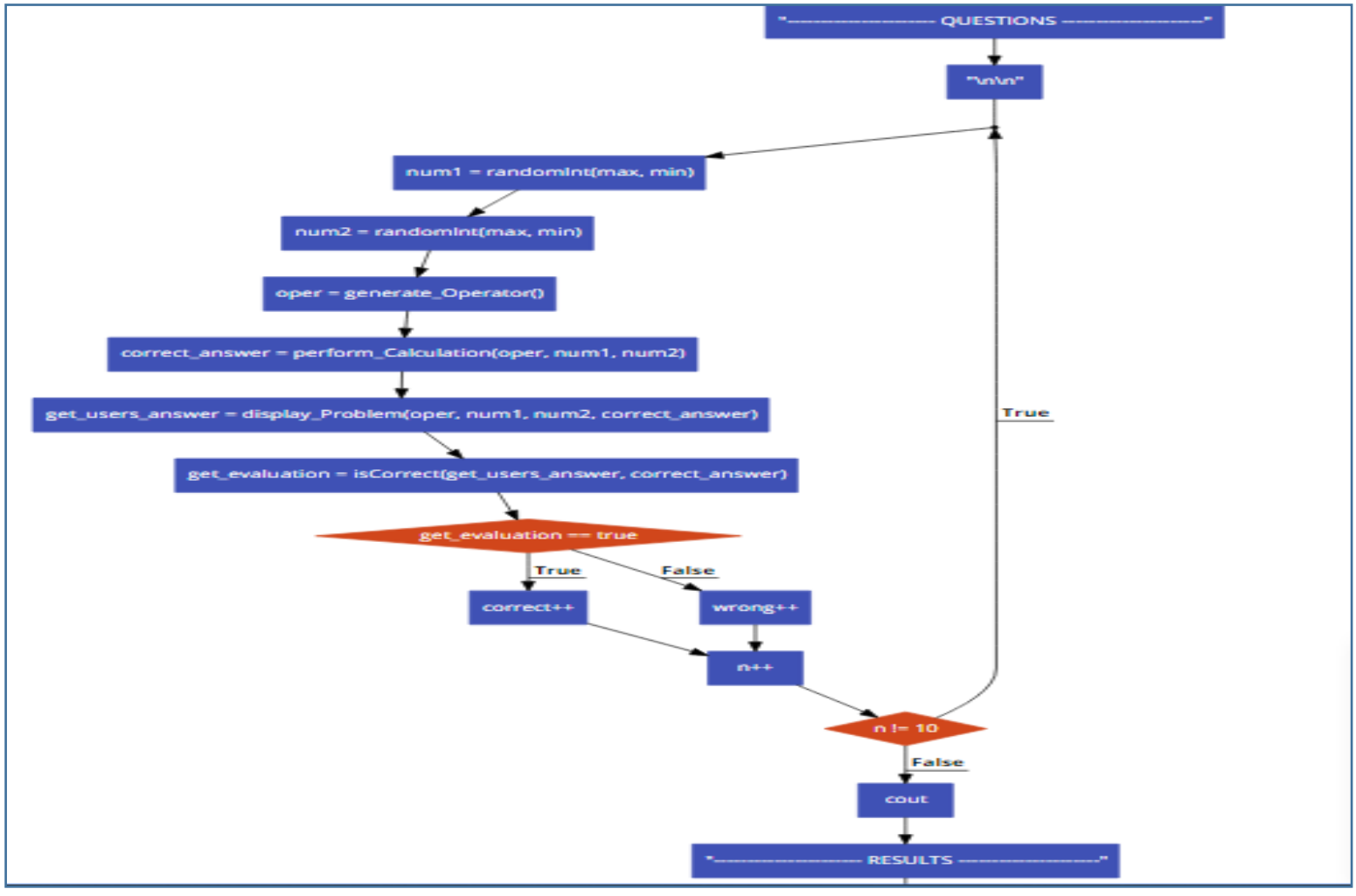
```cpp
bool isCorrect(int users_answer, int correct_answer) {

    if (users_answer == correct_answer) {

        return true;
    }

    return false;


}

void display_Results(int correct, int wrong) {

    cout << "Corrects : " << correct << " " << endl;
    cout << "Wrongs: " << wrong << " " << endl;
}
```

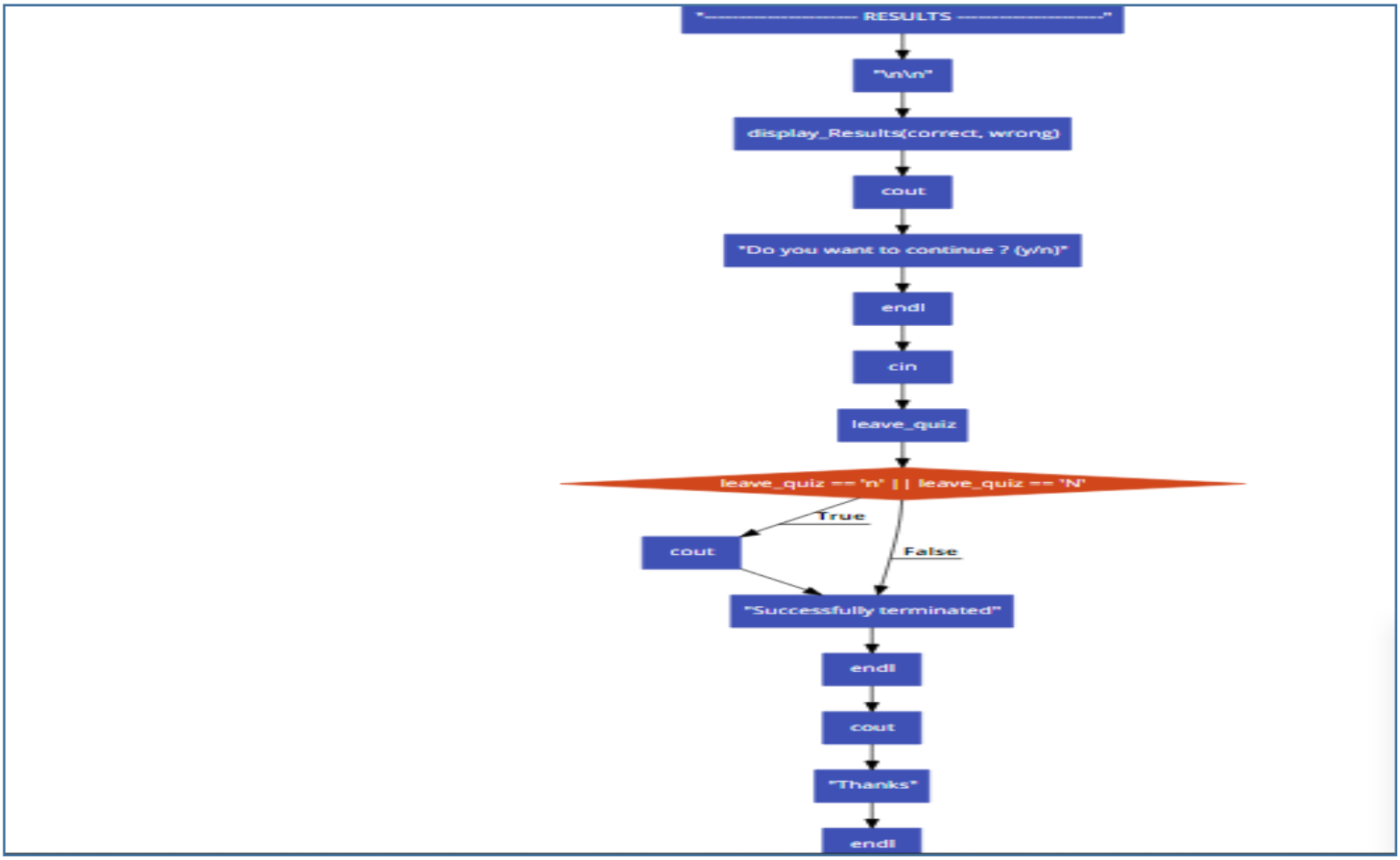

C:\Users\PROJECT VENUS\Documents\Visual Studio 2019\C++ Programming\C++ Projec...
Easy(a)
Moderate(b)
Advanced(c)
Choose: a
---------- QUESTIONS ----------
5-7 = -2
Correct
7-9 = -2
Correct
9+4 = 12
Wrong
Try again
9+4 = 13
Correct
6+9 = 12
Wrong
Try again
6+9 = 12
Wrong
9-1 = 8
Correct
2+6 = 8
Correct
7-7 = 0
Correct
0+1 = 1
Correct
3-5 = -2
Correct
1+5 = 6
Correct
---------- RESULTS ----------
Corrects : 9
Wrongs: 1
Do you want to continue ? (y/n)
y
Levels
Easy(a)
Moderate(b)
Advanced(c)
Choose:

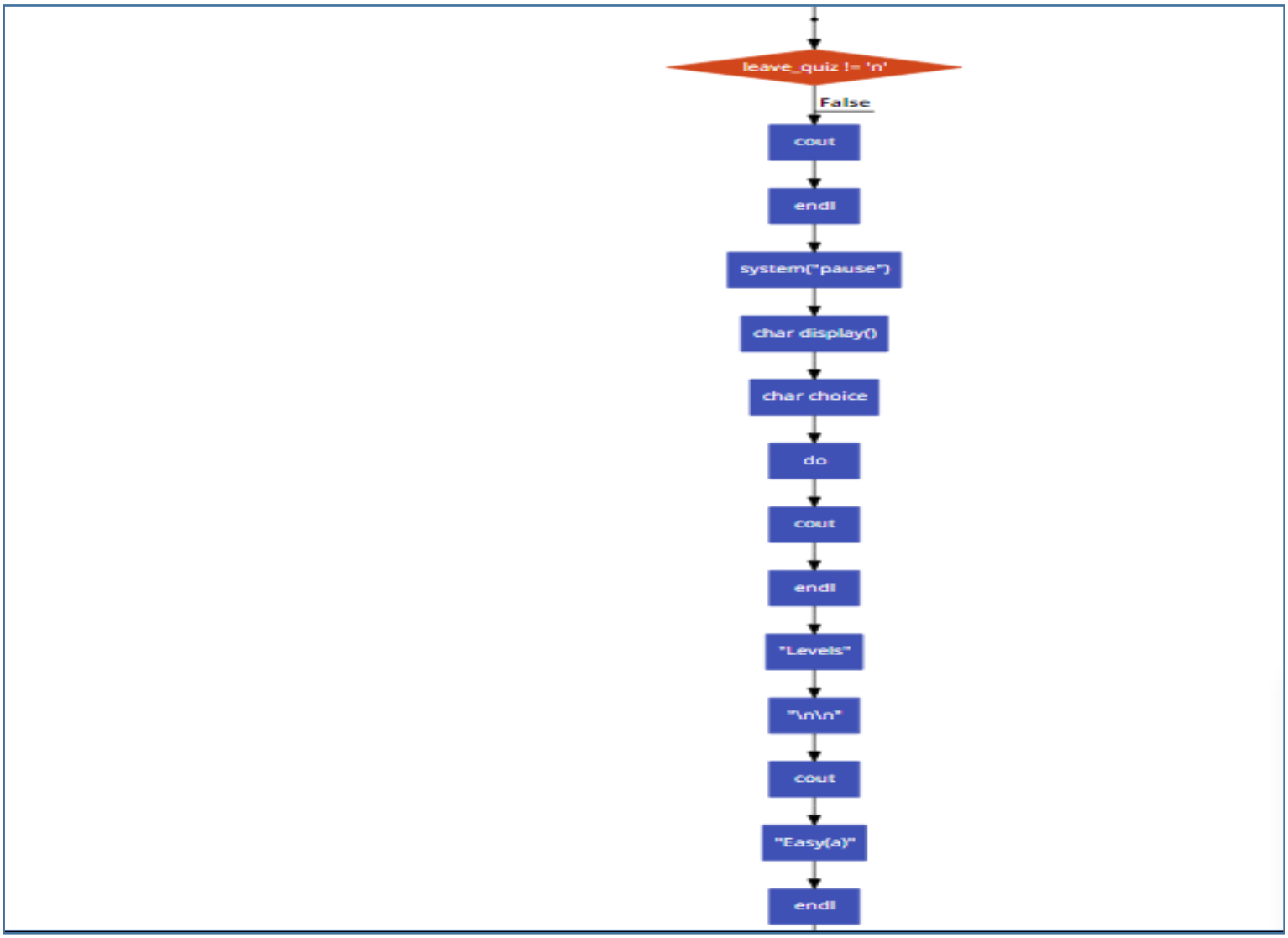
CHAPTER 4

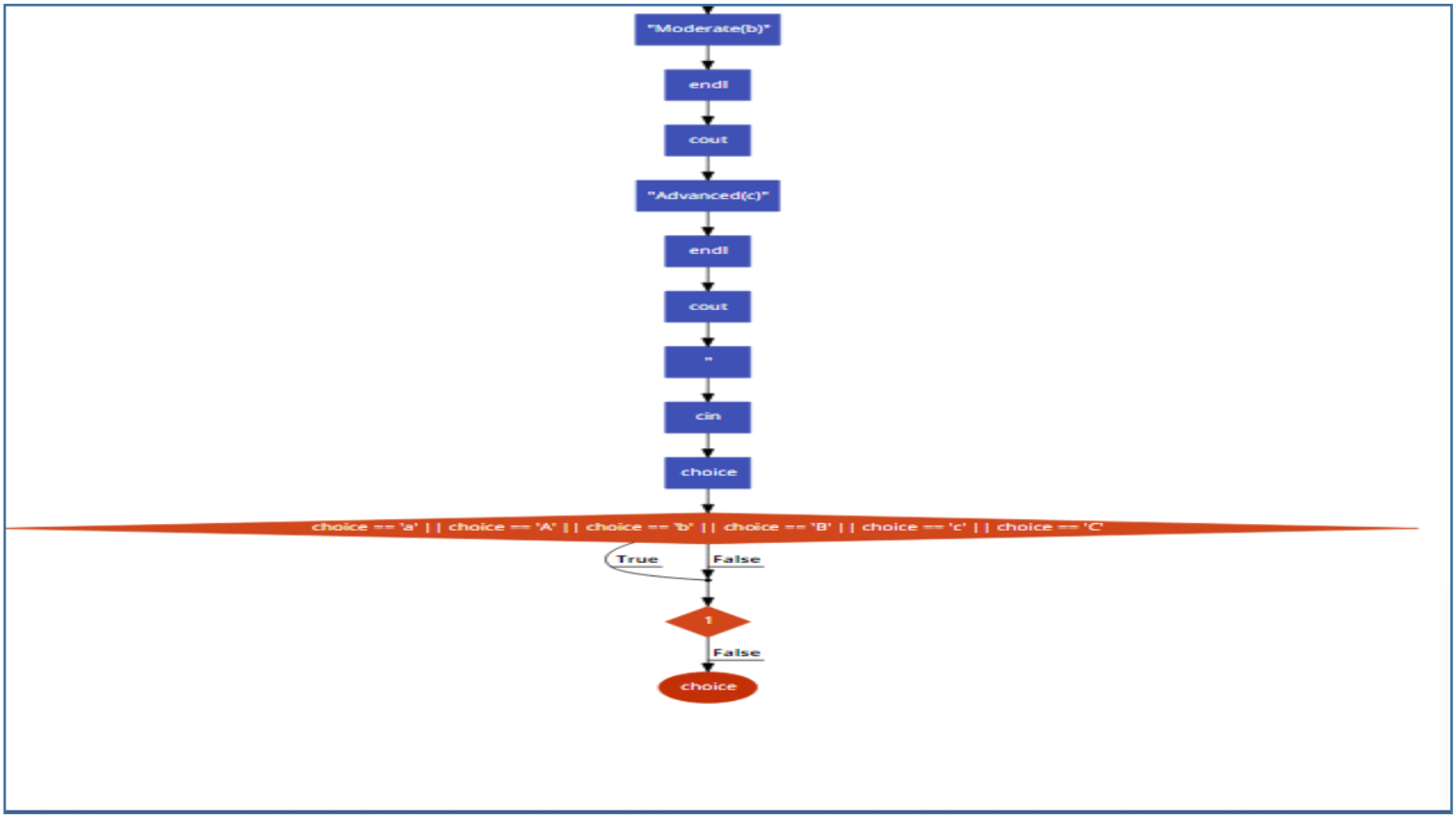
ALGORITHM FLOW

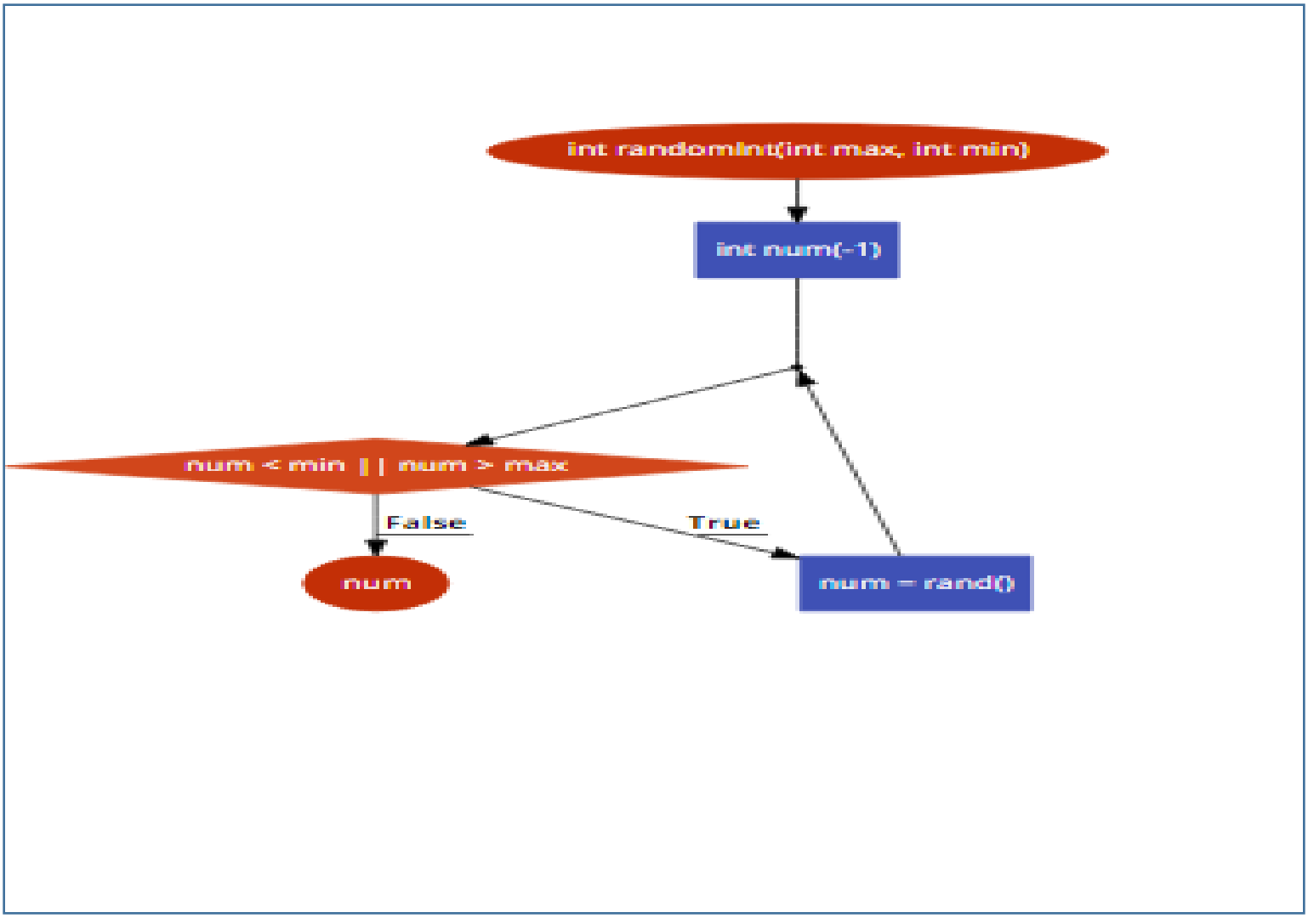

cout
"---------- QUIZ ----------"
"\n\n"
do
level = display()
cout
endl
level == 'a' || level == 'A'
True
False
level == 'b' || level == 'B'
max = 9
True
False
level == 'c' || level == 'C'
max = 999
True
max = 9999
False
min = 0
cout
"---------- QUESTIONS ----------"
"\n\n"

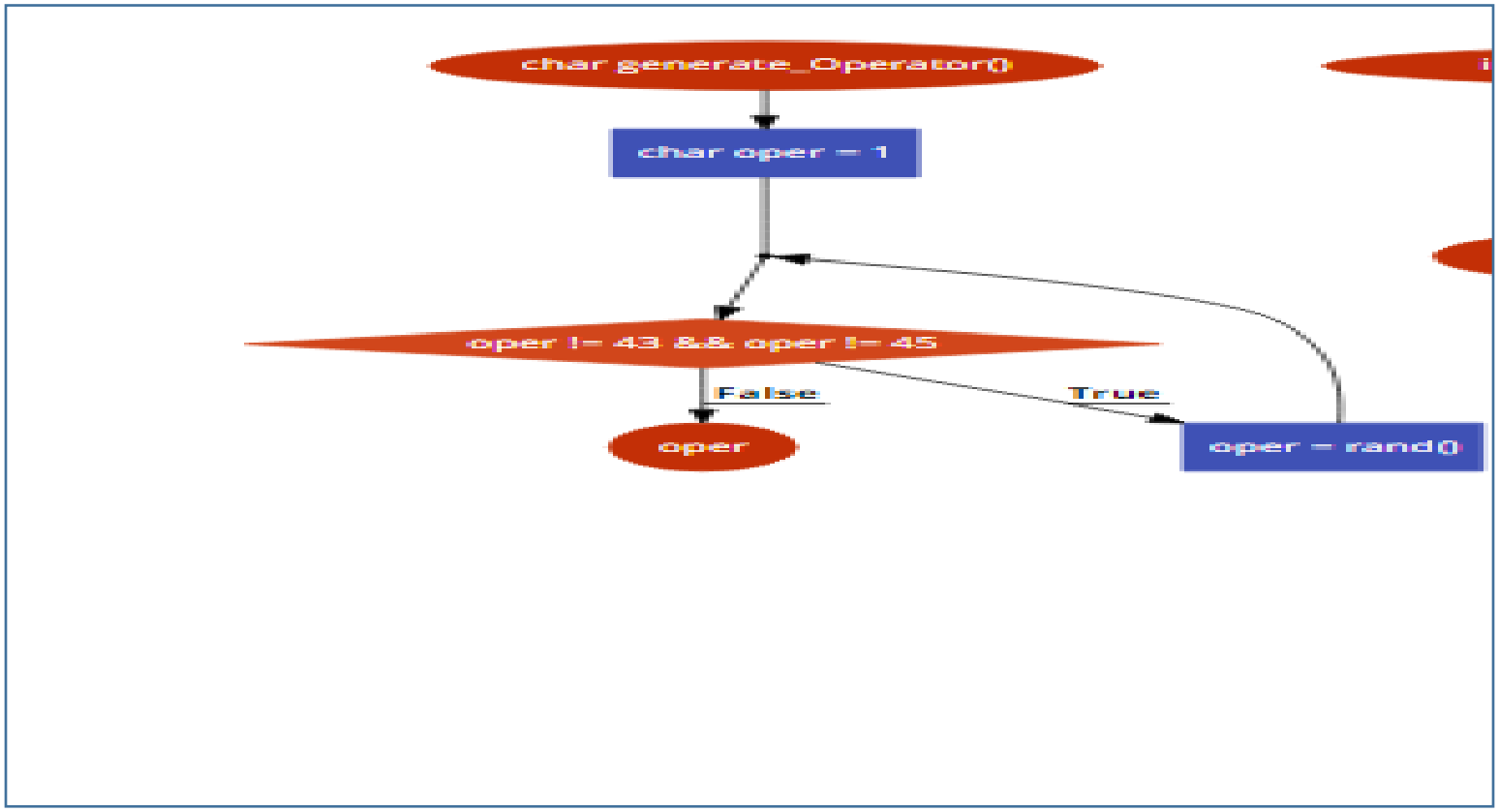

"---------- QUESTIONS ----------"
"\n\n"
num1 = randomInt(max, min)
num2 = randomInt(max, min)
oper = generate_Operator()
correct_answer = perform_Calculation(oper, num1, num2)
get_users_answer = display_Problem(oper, num1, num2, correct_answer)
get_evaluation = isCorrect(get_users_answer, correct_answer)
get_evaluation == true
True
False
correct++
wrong++
n++
n != 10
True
False
cout
"---------- RESULTS ----------"

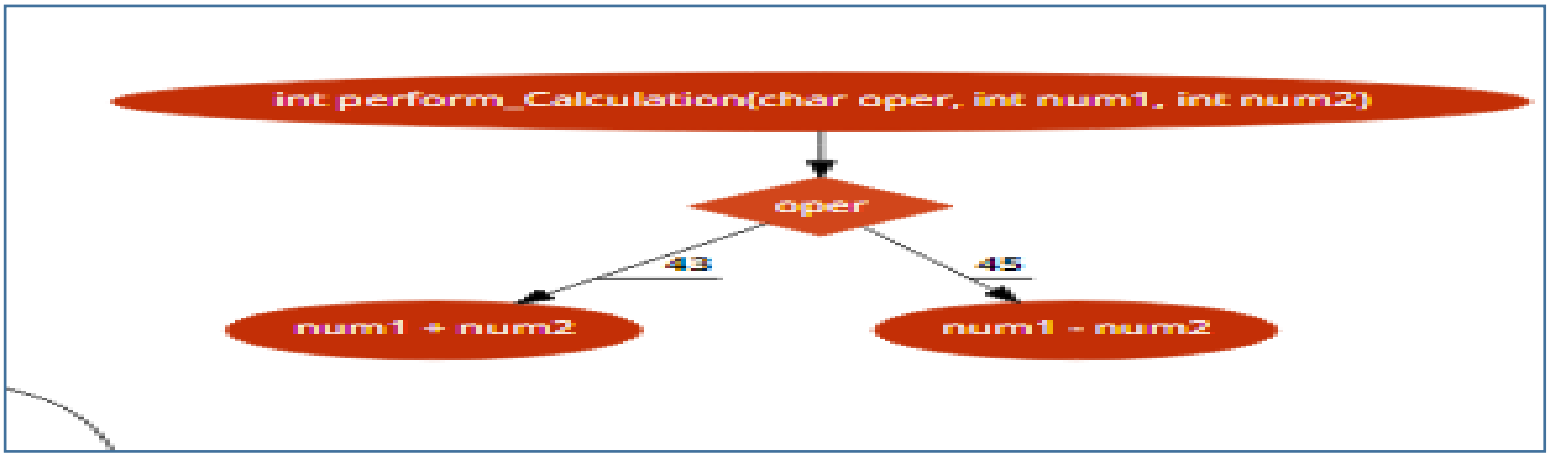

"---------- RESULTS ----------"
"\n\n"
display_Results(correct, wrong)
cout
"Do you want to continue ? (y/n)"
endl
cin
leave_quiz
leave_quiz == 'n' || leave_quiz == 'N'
True
False
cout
"Successfully terminated"
endl
cout
"Thanks"
endl

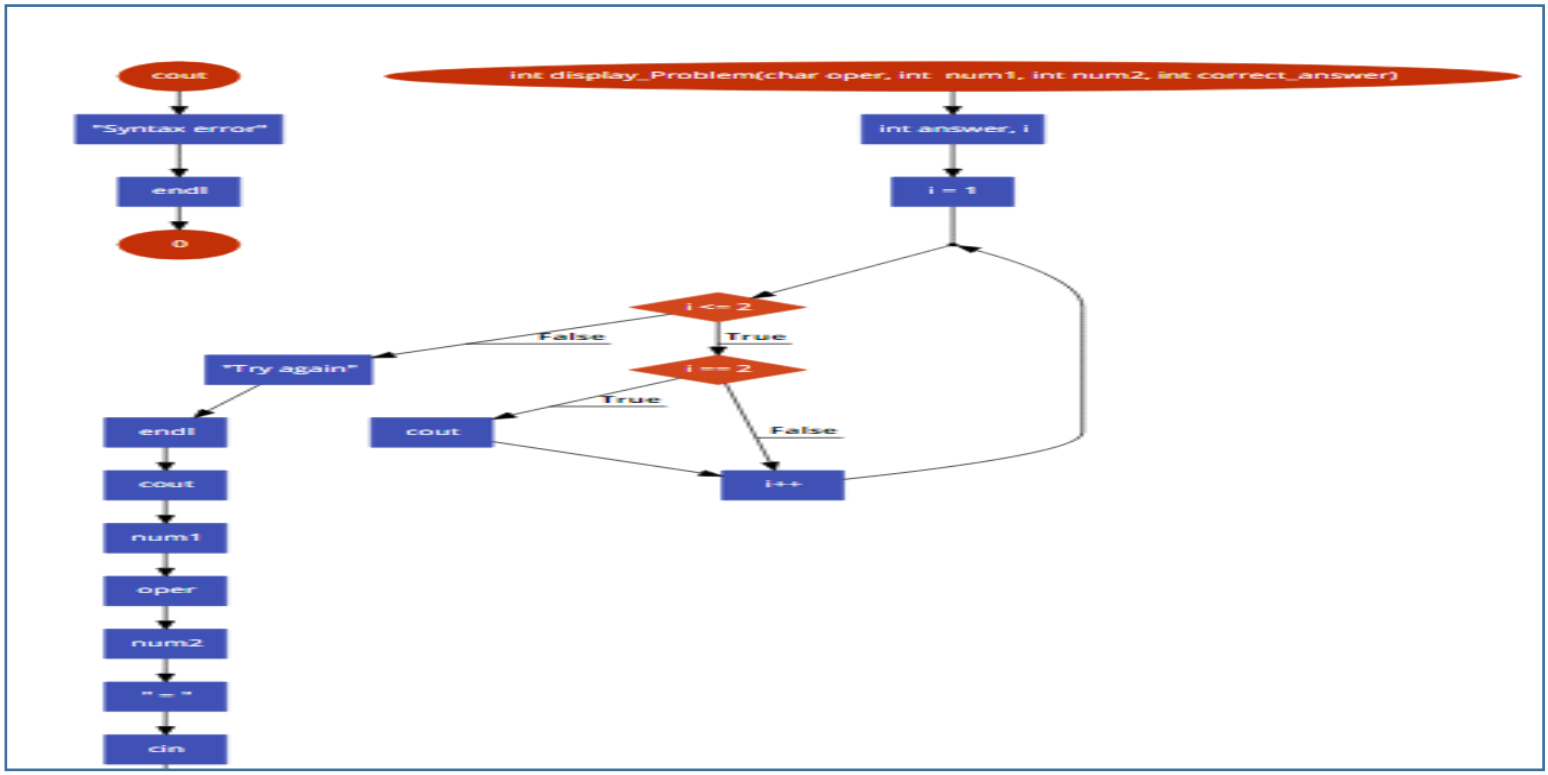

leave_quiz != 'n'
False
cout
endl
system("pause")
char display()
char choice
do
cout
endl
"Levels"
"\n\n"
cout
"Easy(a)"
endl

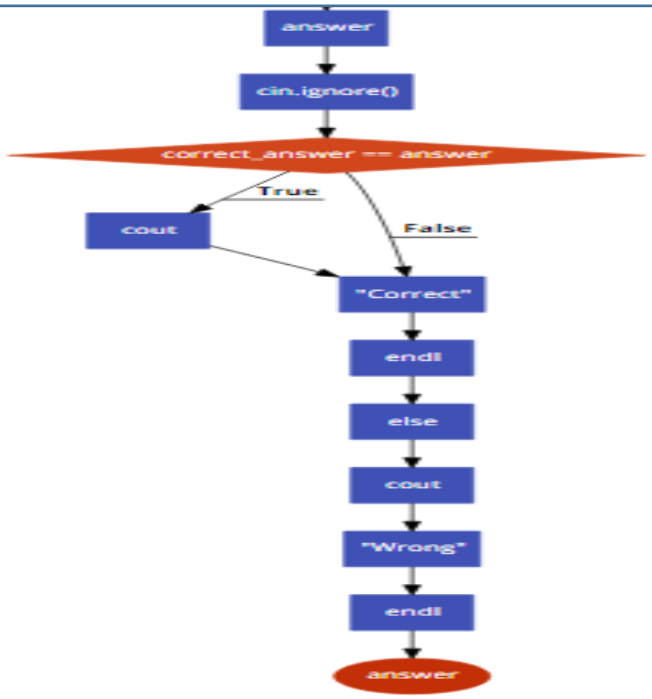

"Moderate(b)"
endl
cout
"Advanced(c)"
endl
cout
"
cin
choice
choice == 'a' || choice == 'A' || choice == 'b' || choice == 'B' || choice == 'c' || choice == 'C'
True
False
1
False
choice

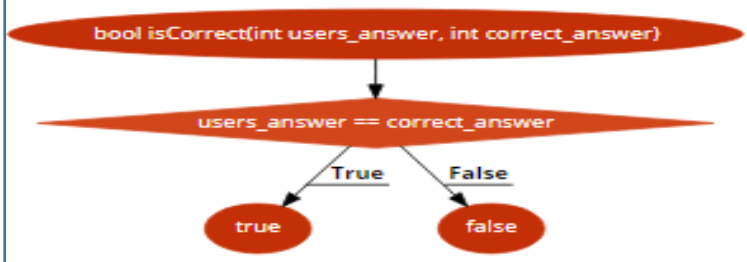

int randomInt(int max, int min)
int num(-1)
num < min || num > max
False
True
num
num = rand()

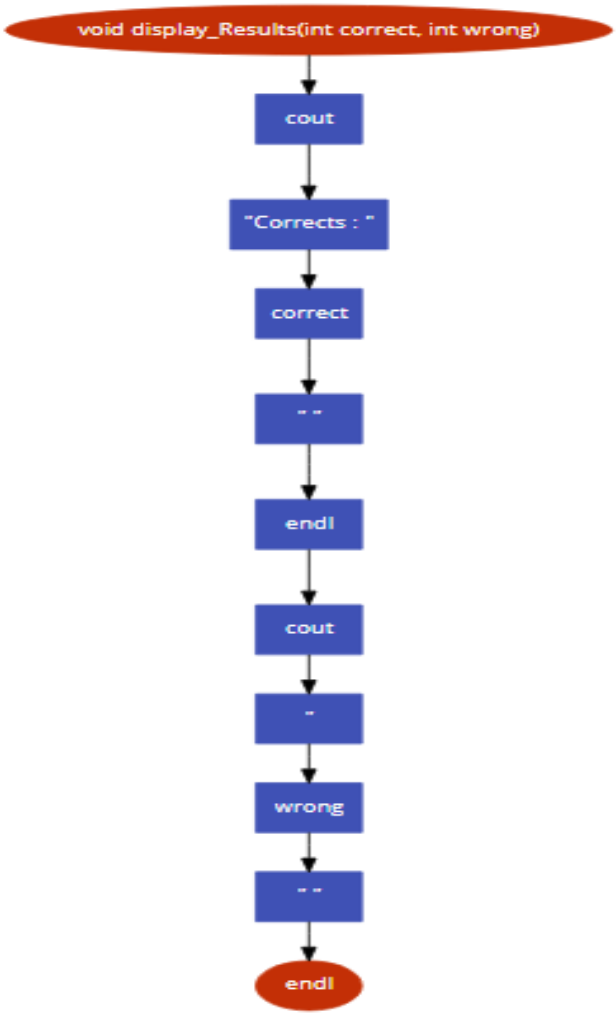

char generate_Operator()
char oper = 1
oper != 43 && oper != 45
False
True
oper
oper = rand()


int perform_Calculation(char oper, int num1, int num2)
oper
43
45
num1 + num2
num1 - num2


cout
"Syntax error"
endl
0
int display_Problem(char oper, int num1, int num2, int correct_answer)
int answer, i
i = 1
i <= 2
False
True
"Try again"
i == 2
True
False
endl
cout
num1
oper
num2
" = "
cin
cout
i++


answer
cin.ignore()
correct_answer == answer
True
False
cout
"Correct"
endl
else
cout
"Wrong"
endl
answer


bool isCorrect(int users_answer, int correct_answer)
users_answer == correct_answer
True
False
true
false


void display_Results(int correct, int wrong)
cout
"Corrects : "
correct
" "
endl
cout
"
wrong
" "
endl