To do list for the week of 2/10/2014

Branching

If statement already covered

If Else and block for today

Also case statement

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accept three grades, find the average

 and display Passed or Failed.

Teach objective - making decisions

By Dr. John Abraham

Created for 1380 students

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <iomanip>

#include <fstream>

using namespace std;

char getgrade (double);//to find letter grade

void printToFile(int, int, int, double, char);

int main ()

{

 int one, two, three;

 double average;

 char grade;

 cout << "Enter three grades ";

 cin >> one >> two >> three;

 average= (one+ two+ three)/3.0;

 cout << "Three grades are: " <<one <<setw(4)<<two <<setw(4)<<three <<endl;

 cout <<fixed<<showpoint<<setprecision(2);

 cout << "The average is : " <<average <<endl;

 grade = getgrade(average);

 cout << "The letter grade is : "<< grade << "\n";

 printToFile(one,two,three, average, grade);

 cin.ignore();

 getchar();

 return (0);

}

char getgrade (double average)

 {

 if (average >=90)return ('A');

 else if (average>= 80) return('B'); else if (average >=70) return ('C');

 else if (average >= 60) return ('D');

 else return('F');

 }

void printToFile(int one, int two, int three, double average, char grade)

{

 ofstream outfile;

 outfile.open("ifThenElse.txt");

 outfile << "Three grades are: " <<one <<setw(4)<<two <<setw(4)<<three <<endl;

 outfile <<fixed<<showpoint<<setprecision(2);

 outfile << "The average is : " <<average <<endl;

 outfile << "The letter grade is : "<< grade << "\n";

 outfile.close();

}

 In the getgrade function if the average is 90 or above the function returns an A and ends

the function; it only continues if the average is not 90 or above. You may want to modify

the program as follows to avoid many return statements. The program reads better when

you only have one return.

char getgrade (double average)

{

 char grade;

 if (average >=90) grade = ‘A’;

 else if (average>= 80) grade = 'B';

 else if (average >=70) grade ='C';

 else if (average >= 60) grade ='D';

 else grade ='F';

 return (grade);

 }

Any time you write a program for multiple alternatives, the program should be run to

check every alternative. In program Run 3-2 only three alternatives are tested. If you

were to turn this program in for a grade, you should include all the alternatives.

Logical operators work with boolean values or results of relational operations. Logical

operators are: AND (&&), OR (||), and NOT (!). For each of this operation we can

obtain a truth table.

T && T => T T || T => T !T => F

T && F => F T || F => T !F =>T

F && F => F F || F => F

Suppose the average score is 95. Let us try this statement:

If (average >=90 && average <= 100)

 cout << “Your grade is A \n”;

The first relational operation of average >= 90 yields a T. The second operation of

average <= 100 also yields a T. T && T gives a T. Since the entire operation yields a

true then “Your grade is A” is displayed. If, on the other hand, the average score is 88,

the first operation will yield a false and the second operation will yield a true (88 is less

than 100); F && T is False, and the output will not be displayed. The concept of logical

and relational operators will become clearer in the next chapter when we deal with

repetitions. -------------------------- Switch Case--

Suppose you created a menu to choose one of the items from a list you may have to write

some thing like this:

If (choice==1) AddClient ();

else if (choice == 2) EditClientInfo ();

 else if (choice==3) LookUpClient();

and so on…

If the menu has many items there is a lot of coding you have to do and the code is hard to

follow. There is alternative to the multiple if/else statements. We can use the switch

statement as shown below.

Switch (choice)

{

 case 1: AddClient();

 break;

 case 2: EditClientInfo();

 break; case 3: LookUpCleint();

 break;

 and so on..

 default :

 cout << “invalid response”;

}

 We are essentially telling c++ to do something in case of choice is 1, 2, or 3 and

so on. If a match is not found then it falls to the default and carries out that instruction.

You need to remember that you cannot use any relational operators with the case

statement such as case >3. What happens if you do not include the break statements?

Every case statement will be executed until it finds the break statement or until the end of

the block. Try deleting the break statements and see what happens.

Here is a complete example of a program. Let us write a program to recieve three grades,

find its average, determine the letter grade and write a brief comment about the grade.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Accept three grades, find the average

 and display Passed or Failed.

Teach objective - making decisions

By Dr. John Abraham

Created for 1380 students

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <iomanip> //to format input and output. Here setw and endl require it

#include <fstream>

using namespace std;

char getgrade (double);//to find letter grade

void Message (int, int, int, double, char);

void printToFile(int, int, int, double, char);

int main ()

{

 int one, two, three;

 double average;

 char grade;

 cout << "Enter three grades ";

 cin >> one >> two >> three;

 average= (one+ two+ three)/3.0; // the total of three grades are first converted

to float

 grade = getgrade(average);

 Message (one,two,three,average,grade);

 printToFile(one,two,three, average, grade);

 cin.ignore();

 getchar();

 return (0);

 }

char getgrade (double average)

 {

 if (average >=90)return ('A');

 else if (average>= 80) return('B');

 else if (average >=70) return ('C');

 else if (average >= 60) return ('D');

 else return('F');

 }

void Message (int one, int two, int three, double average, char grade)

 {

 cout << "Three grades are: " <<one <<setw(4)<<two <<setw(4)<<three <<endl;

 cout <<fixed<<showpoint<<setprecision(2);

 cout << "The average is : " <<average <<endl;

 cout << "\nThe letter grade is : "<< grade << endl;

 switch (grade)

 {

 case 'A' : cout << "Very impressive grade indeed!\n";break;

 case 'B' : cout << "A solid performance, congratulations!\n"; break;

 case 'C' : cout << "C++ is a tough course, but YOU MADE IT!\n";break;

 case 'D' : cout << "Made it eh? \n";break;

 case 'F' : cout << "Don't give up. Try keeping up with all the homework!\n";

 }

 }

void printToFile(int one, int two, int three, double average, char grade)

{

 ofstream outfile;

 outfile.open("ifThenElse.txt");

 outfile << "Three grades are: " <<one <<setw(4)<<two <<setw(4)<<three <<endl;

 outfile <<fixed<<showpoint<<setprecision(2);

 outfile << "The average is : " <<average <<endl;

 outfile << "The letter grade is : "<< grade << "\n";

switch (grade)

 {

 case 'A' : outfile << "Very impressive grade indeed!\n";break;

 case 'B' : outfile << "A solid performance, congratulations!\n"; break;

 case 'C' : outfile << "C++ is a tough course, but YOU MADE IT!\n";break;

 case 'D' : outfile << "Made it eh? \n";break;

 case 'F' : outfile << "Don't give up. Try keeping up with all the homework!\n";

 }

 outfile.close();

}

Introduction to loop

While loop

Here are the steps:

1. Choose a Loop Control Variable.

1a. Initialize the variable (this variable is also called the loop control variable or

LCV, because this variable controls the loop). In this example the LCV is

number.

2. Check for the condition to enter the loop. The condition should yield a True

to enter the loop. If the condition yields a false, the loop is not entered.

3. Set up the body of the loop. You may have one or multiple things to do

within the loop body. The body here appears within the brackets.

4. Change the value of the LCV within the body. In this example the number is

changed by adding a one to it

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Display 1 to 100 on the screen

Teaching objective - while loop

By Dr. John Abraham

Created for 1370 students \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

using namespace std;

int main()

{

 int number;

 number = 1;

 while (number <= 100)

 {

 cout << number << " "; //display number and put some spaces or use setw()

 number ++; //increment number by one

 }

 getchar();

 return (0);

}

This while loop is a count controlled loop, since we wanted to repeat the loop a certain

number of times. The count controlled loop may also be referred to as a step controlled

loop. We can write a sentinel controlled loop as well. A sentinel value is a value that

belongs to a type but does not belong to the set you are working with. For example

suppose you are entering names. Names belong to a type called string. When asked for

the name what if you entered ‘quit’?. ‘Quit’ also belongs to the string type, however,

does not belong to the set of names.