

Lab 5 - CMPS 1043- Computer Science I

If\Else Statements & Nested If/Else Statements

Objectives: Demonstrate use of decision operator IF in a variety of problem scenarios.
Demonstrate use of relational operators.

Relational Operator (pg. 158): Operator that allows comparison of values to determine

<	less than	>	greater than
<=	less than or equal to	>=	greater than or equal to
==	equal to	!=	not equal to

The result of a relational operator is a Boolean value, *true* or *false*.

Decision Statements: Statements that allow execution under various circumstances. `if` is one of the Decision Statements. (Pg. 173)

```
if (condition)
{
    Statement set 1;
}
else
{
    Statement set 2;
}
```

When *condition* represents a true statement, Statement Set 1 will be executed.

When *condition* represents a false statement, Statement Set 2 will be executed.

Open a new project and type in the following example:

```
int X = 10, Y = 4;
if (X <= Y)
{
    cout << "X =" << X << "Y =" << Y << "\n";
    cout << "X is smaller than or equal to Y \n";
}
else
{
    cout << "X =" << X << "Y =" << Y << "\n";
    cout << "X is greater than Y \n";
}
```

Change the values of X & Y and run.

Note: When a statement is in BOTH the `if` and the `else` portion of the statement, it can be moved. Move the first `cout` statement to the line immediately above the `if` statement and execute again.

if/else if statement (pg. 175) - used to select among a set of conditions until one is found to be true.

```
if (condition 1)
{
    Statement set 1;
}
else (condition 2)
{
    Statement set 2;
}
...
else (condition n)
{
    Statement set n;
}
```

Modify your program as shown here and run for different values of X & Y.

```
int X = 10, Y = 4;
cout << "X =" << X << "Y =" << Y << '\n';
if (X < Y)
{
    cout << "X is less than Y \n";
}
else if (X > Y)
{
    cout << "X is greater than Y \n";
}
else
{
    cout << "X is equal to Y \n";
}
```

Note this strategy can be used even when more than one of the conditions can be true. In this case it is important to put the statements in the correct order. Type in and execute the following example which determines the letter grade based on the numeric grade. Try a variety of values for Score.

```
int Score;
cout << "Enter the student's test score\n";
cin >> Score;
cout << "The student's test score is " << Score << '\n';
if (Score >= 90)
{
    cout << "The letter grade is A\n";
}
else if (Score >= 80)
{
    cout << "The letter grade is B\n";
}
```

```

}
else if (Score >= 70)
{
    cout << "The letter grade is C\n";
}
else if (Score >= 60)
{
    cout << "The letter grade is D\n";
}
else
{
    cout << "The letter grade is F\n";
}
}

```

Why is there no `if` (condition on) the final `else` statement? Would it still work if `if (Score < 60)` is added?

What happens if you take the "else" off each of the statements and simply do a series of `if` statements? Try it.

Assignment: Develop and execute a program to solve the following problem. Print & turn in code and solutions: Andrew's Department Store is having a BIG sale on everything in the store. Customers will receive additional discounts on their total purchase depending upon how much they buy. Write a program that asks for the total purchase amount then calculates the discount and new amount, then prints out all 3 values as shown below. The discounts are to be as follows.

<u>Total Purchase</u>	<u>Discount</u>
Less than \$100	10%
\$100 to \$249.99	20%
\$250 to \$499.99	30%
\$500 to \$999.99	40%
\$1000 or more	50%

Output should look as follows:

```

Total Purchase    $ 1234.00
Discount          $   617.00
-----
New Price         $   617.00

```

Include the following `while` loop (as in earlier programs) and enter the data items:

```
1234 48.98 378.98 200.00 783.99 2.08
```

```

count = 1
while (count <= 6)
{
    \\ Insert cin & calculations here - output to file
    count +=1;
}

```