In this module we will be covering new statements: if, if-else, if-elif-else; we also will cover logical operators, and boolean variables

Many times in a computer program we would like to put several programming statements together and make a decision in those statements based on the data presented.

A **"control structure"** is a logical design of statements that controls the order in which a set of statements execute. A **sequence structure** is a set of statements that execute in the order that they appear. A **decision structure** is one where you can only execute a set of statements under certain conditions (provided through the data) - we also call this a **selection structure.**

In your flowchart link in the last module - you saw flowcharts that used a diamond symbol that could show multiple actions based on the data provided. You may have one alternative path other than sequential, or multiple paths for executing your statements. Pages 82 and 83 of your text show flowchart examples for this logic.

Again - this is LOGIC. This is what you would understand you need to do based on your "designing the program" phase!!

Boolean Operators

We use Boolean operators with our control structures to "test" the data given to see which path we want to follow in our decision structure (selection structure).

The operators are:

> greater than

< less than

>= greater than or equal to

<= less than or equal to

== equal to

!= not equal to

IF Statements

The SYNTAX for an if statement is:

if *condition:*

statement

statement

etc.

The *if* statement tests an expression to determine whether it is true or false. If the expression is true (based on the data), then the *statement(s)* following are executed. If the *if* statement is false, then the *statements* that are part of the *if* statement are not executed and you "jump" to the statement following the last statement in the *if* block.

IF-Else Statements

This statement has two different sets of statement(s) to execute whether the statement is true (one block of statements) or false (another block of statements). The SYNTAX is:

if *condition:*

statement

statement

etc.

else:

statement

statement

etc.

If the condition is true the first set of statements is executed, if the condition is false the second set of statements is executed.

If you use *if* statements to compare strings - each string is compared letter by letter. The ASCII and UNICODE (see links 2 & 3) character set show you which letters and characters are greater than the others.

Nested Decision Structures (nested IF statements)

We can actually nest one decision structure (if statement) inside of another one. Pages 97-99 show you some flowcharts with the logic involved with this type of statement. You can actually do multiple nesting of *if* statements. We recommend, however, that you don't nest things deeper than 3.

if-elif-else Statement

This statement takes the place of multiple nested *if* statements. The SYNTAX is:

if *condition_1:*

statement

statement

etc.

elif *condition_2:*

statement

statement

etc.

else:

statement

statement

etc.

Logical Operators

There are 3 basic logical operators: *and, or, NOT*

If you remember your math - we look at using Truth Tables for evaluating AND, OR and NOT

For AND:

true and false = false

false and true = false

false and false = false

true and true = true

For OR:

true or false = true

false or true = true

false or false = false

true or true = true

For NOT:

not true = false

not false = true

Boolean Variables

There are only two values for a Boolean data type variable - True and False.