Selection Structure

- A selection structure is used in a program to execute a command or function only if a certain condition is true.
- If the condition is false, another command or function is executed (or nothing is done at all).

Condition

A condition is a logical comparison statement which compares two items using a logic operator and returns either **True** or **False**.

The **logic operators** are:

1. == (equal to)
2. > (greater than)
3. < (less than)
4. >= (greater than or equal to)
5. <= (less than or equal to)
6. != (not equal to)

In programming, conditions are usually used check the value of a variable.

Selection Structures in Python

1. Binary Selection Structure

The most commonly used selective structure is the binary selective structure. This structure evaluates (checks) a condition and then performs one or more commands if the condition is true **OR** one or more commands if the condition is false.

```
# Selective Structure Example 1: A binary selection structure

age = int (input ( "How old are you?")) # User's age

if age >= 16:
        print ( "You are old enough to drive.")
else:
        print ( "You are not old enough to drive.")
```

The new commands are described below:

if age >= 16:
- The **if** command starts the selective structure.
- **age >= 16** is the condition. This statement checks to see if the condition is true.
- **if** statement must end in a colon (:)

print("You are ...")
- This **print** function is executed (done) only if the condition is true.
- Note any code in a selection structure must be indented. This is how Python indicates that a command is to be done if the

condition is true. The easiest way to do it is to use the Tab key

else: - The print function below is executed (done) only if the condition is false.
- Again, it must end in a colon

print ("You are not **..)** - This line is outputted if the other put statements are not outputted.
- Again, it must be indented.

2. Simple Selection Structure

This structure evaluates (checks) a condition and then performs one or more commands if the condition is true. It does nothing if the condition is false.

```
# Selective Structure Example #2: A simple selection structure

answer = input ( "Is it sunny outside?")

if answer == "yes":
    print ("Put on suncreen!")      # Only done if user inputs yes

print ("Program done!")             # This line is always done because it is not indented and
                                    # therefore not part of the selection structure.
```

3. Multiple Selection Structure

This structure evaluates(checks) more than one condition.

- The computer evaluates the 1st condition and if the condition is true, it performs one or more commands and then exits the selection structure.
- If the 1st condition is not true, the computer evaluates the 2nd condition. If the 2nd condition is true, it performs one or more commands and then exits the selection structure.
- This continues until the **else** statement. The commands following the else statement are executed if none of the preceding conditions are true.

```
# Selective Structure Example 3: A multiple selection structure with equal operators
print ( "Which of the following sports do you like the best?")
print ( "hockey, baseball, swimming, tennis")
sport = input()             #There is no prompt on this line since the user is prompted above''
if sport == "hockey" :
    print ("Then you must watch Hockey Night In Canada.")
elif sport == "baseball" :                          # Note the spelling of elif!
    print ("Then you must have watched a game or two at Skydome.")
elif sport == "swimming" :
    print ("Then you must enjoy the Olympics since there are a lot of aquatics.")
elif sport == "tennis" :
    print ("Then you must have watched Wimbledon.")
else :
    print ("Your favourite sport is not on the list")
```

```
''' Selective Structure Example 4:
A multiple selection structure with a range of numbers. '''
mark = int (input ("What mark do you hope to get in this course?"))
if mark >= 80 :
        print ("Keep up the good work!")
elif mark >= 60 :
        print ("Satisfactory, but you can do better.")
elif mark >= 50 :
        print ("You better work harder")
else :
        print ("You are failing!")
        print ("Make an appointment with your Guidance Counsellor!")
```

Note the order of the conditions. Why would this selective structure not work if we had started with the condition *if mark >= 50*?