MODULE 1

PPT -6

Python input() Function

Python provides us with two inbuilt functions to read the input from the keyboard.

1. **raw_input (prompt)# for older version (2.X)**
2. **input (prompt)**

```
print("Enter your name:")
x = input()
print("Hello, " + x)
```

Control structures

- A control structure (or flow of control) is a block of programming that analyses variables and chooses a direction in which to go based on given parameters.
- A program's control flow is the order in which the program's code executes.
- So, it is the basic **decision-making process** in programming and flow of control determines how a computer program will respond when given certain conditions and parameters.
- **Flow of control** through any given program is implemented with three basic types of control structures: Sequential, Selection and Repetition.

- **Sequential**

Sequential execution is when statements are executed one after another in order. You don't need to do anything more for this to happen.

- **Selection**

Selection used for decisions, branching - choosing between 2 or more alternative paths.

1. if
2. if...else
3. switch

- **Repetition-**

Repetition used for looping, i.e. repeating a piece of code multiple times in a row.

1. while loop
2. do..while loop
3. for loop

The if Statement

- Often, you need to execute some statements only if some condition holds, or choose statements to execute depending on several mutually exclusive conditions. The Python compound statement if, which uses if, elif, and else clauses, lets you conditionally execute blocks of statements.

if *expression*:
 statement(s)
elif *expression*:
 statement(s)
elif *expression*:
 statement(s)
 ...
else:
 statement(s)

Note- The elif and else clauses are optional. Note that unlike some languages, Python does not have a switch statement, so you must use if, elif, and else for all conditional processing.

Equals: a == b

Not Equals: a != b

Less than: a < b

Less than or equal to: a <= b

Greater than: a > b

Greater than or equal to: a >= b

- These conditions can be used in several ways, most commonly in "if statements" and loops.

```
a = 200
b = 33
if b > a:
  print("b is greater than a")
else:
  print("b is not greater than a")
```

- The elif keyword is pythons way of saying "if the previous conditions were not true, then try this condition".
- The else keyword catches anything which isn't caught by the preceding conditions.
- You can also have an else without the elif.
- If you have only one statement to execute, you can put it on the same line as the if statement.

```
a = 200
b = 33
if a > b: print("a is greater than b")
```

If you have only one statement to execute, one for if, and one for else, you can put it all on the same line:

```
a = 2
b = 330
print("A") if a > b else print("B")
```

- You can also have multiple else statements on the same line:

```
a = 330
b = 330
print("A") if a > b else print("=") if a == b else print("B")
```

- The and keyword is a logical operator, and is used to combine conditional statements:

```
b = 33
c = 500
if a > b and c > a:
  print("Both conditions are True")
if a > b or a > c:
  print("At least one of the conditions is True")
```

Nested If

- You can have if statements inside if statements, this is called *nested* if statements.

```
x = 41
if x > 10:
  print("Above ten,")
  if x > 20:
    print("and also above 20!")
  else:
    print("but not above 20.")
```

The pass Statement

- if statements cannot be empty, but if you for some reason have an if statement with no content, put in the pass statement to avoid getting an error.
- Example
- ```
  a = 33
  b = 200
  if b > a:
    pass
  ```
  ```

Programs based on if statement

1- Write a program to input an integer and check

If number if negative integer then print number

If non-negative number

- Check if number is odd, print odd number
- Else: number is even, print even number