PYTHON NOTES

1. What is Python?

- **Python** is a high-level language like other high-level language such as Java, C++, PHP, Ruby, Basic and Perl.
- Python is an object-oriented programming language.
- Python provides security.
- The CPU understands a language which is called as **Machine Language**.
- Machine language is very complex and very troublesome to write because it is represented all in zero's and one's.
- The actual hardware inside CPU does not understand any of these high-level languages.

2. Program:

- A **Program** can be defined as a set of instructions given to a computer to achieve any objective.
- Instructions can be given to a computer by writing programs.
- Tasks can be automated by giving instructions to the computers.

3. **Defining Computer Hardware**:

Components:

- **CPU**: It helps in processing the instructions.
- **Main Memory**: It provides storage support during execution of any program in computer Eg: RAM.
- **The Secondary Memory**: It helps to store the data permanently inside the computer. Eg: Disk drives, flash memory, DVD and CD.
- **The Input and Output Devices**:
  - **Input Devices** helps users to generate any command or input any data.
  - **Output Device** helps user to get output from computer. Eg: Mouse, Printer, Keyboard, Monitor etc.

4. **Constants and Variables**:

- **Variables** can have any name, but Python reserved words cannot be used.
- A variable provides a named storage that the program can manipulate.

- Variables are named memory location used to store data in program which keeps on changing during execution.
- Programmers can decide the names of the variables.
- Fixed values used in programs such as numbers, letters and strings are called "**Constants"**.
- Values of constants never change during program execution.

5. **Variable Naming Conventions**:

- Must start with a letter or an underscore "_".
- Must consist of a letters, numbers and underscores.
- It is a case sensitive.
- Eg: First_Name, Age, Num1.
- Cannot be used: 1_hello, @hello, h123#2, -abc.
- **Note**: You cannot use reserved words for variable names and identifiers.

6. **Mnemonic Variable Names**:

- Use simple rules of variable naming and avoid reserved words.
- While using simple rules, we have a lot of choice for variable naming.

- Initially this choice can be confusing either in reading or writing the program.
- The following two programs are identical in terms of what they accomplish, but very different when you read and try to understand them:

Eg 1: a=35.0

```
b=12.50
c=a*b
print(c)
```

O/P: 437.5

Eg 2: hours=35.0
rate=12.50
pay=hours*rate
print(pay)
O/P: 437.5

7. **Reserved Words in Python**:

- and, as, assert, break, class, continue, def, del, elif, else, except, exec, finally, for, from, global, if, import, in, is , lambda, not, or, pass, print, raise, return, try, while, with, yield.

8. Compilers and Interpreters:

- **Compiler** is a computer program(or a set of programs) that transforms source code written in a programming language into another computer language.
- **Interpreters** reads the source code of the program, line by line, passes the source code, and interprets the instructions.

9. Python language:

- The **Python language** acts as an intermediator between the end user and the programmer.
- Python script will have .py extensions.
- Every one line can be a program in Python.

10. Types of errors:

- **A syntax error**: It occurs when the "grammar" rules of Python are violated.
- **A logic error**: It occurs when the program has good syntax but there is a mistake in the order of the statements.

  Eg:

  - Using wrong variable name.

- Making a mistake in a Boolean expression.
- Indenting a block to the wrong level.
- Using integer division instead of floating-point division.

- **A Semantic error**: It occurs when the description of the steps to take is syntactically perfect, but the program does not do what it was intended to do.

11. **Difference between Programmers and Users**:

- **Programmers** use software development tools available in a computer to develop software for the computer.
- A programmer may write the program to automate the task for himself or for any other client.
- After learning programming language, the programmer can develop the software that can be utilized by end users.
- **Users** use the tools available in a computer like word processor, spreadsheet etc., whereas **programmers** learn the computer language and develop these tools.

12. Building blocks of a Program:

These are some of the conceptual patterns that are used to construct a program:

- **Input**: Input will come from the user typing data on the keyboard.
- **Output**: Display the results of the program on a screen or store them in a file.
- **Sequential Execution**: Perform statements one after another in the order in which they are encountered in the script.
- **Conditional Execution**: Checks for certain conditions and then execute or skip a sequence of statements.
- **Repeated Execution**: Perform some set of statements repeatedly, usually with some variation.
- **Reuse**: Write a set of instructions once then reuse those instructions in the program.

13. Various Components of programming statements:

- Variable.
- Operator.
- Constant.

- Reserved Words.

14. **Operators and its Precedence**:

- **Operators** are used to manipulate the values of operands.
- There are various types of operators used in program:
    - Comparison (relational) operators.
    - Assignment operators.
    - Logical operators.

15. **Arithmetic Operators**:

- Are the symbols that are used to perform arithmetic operations on operands.

Types of Arithmetic operators:

- + ,- ,* ,/ ,%.

16. **Comparison Operators**:

- Compares the values of an operands and decide the relation among them.
- They are also called as **Relational Operators**.

Types of Comparison Operators:

- < - less than.
- >- greater than.
- <= - less than equal to.
- >= - greater than equal to.
- == - equal to.
- != - not equal to.

17. **Logical Operators**:

- Are used to evaluate expressions and return a Boolean value.

Types of Logical Operators:

- **x && y**: Performs a logical AND of the two operands.
- **x || y**: Performs a logical OR of the two operands.
- **! x**: Performs a logical NOT of the operand.

18. **Logical Operators (Contd..):**

- There are three logical operators and, or and not.

- The semantics of these operators is similar to their meaning in English.
- **Eg**: x>0 and x<10 (is true only if x is greater than 0 and less than 10).
- n %2==0 or n % 3==0(is true if either of the condition is true).
- **The not operator** negates a Boolean expression.
- **Eg**: not(x>y) is true if x>y is false.

19. **Operator Precedence**:

- When we use multiple operators in an expression, program must know which operator to execute first. This is called as **"Operator Precedence"**.
- The following expression multiple operators but they will execute as per precedence rule:
    - X=1+2*3-4/5**6.
- **Eg 1**: b=10, a=5, b%a O/P: 0.
- **Eg 2**: b=10, a=5, b%a==5 O/P: False.

20. **Highest Precedence rule to Lowest Precedence rule**:

- Parentheses are always respected hence given first priority.
- Exponentiation (raise to a power).