Unit - I

Problem solving, programming & Python programing

General problem solving Concept
- → problem solving in everyday life
- → types of problem
- → problem solving with computer
- → difficulties with problem solvin
- → problem solving aspects
- top down design

problem solving in everyday life:-

People make a decision everyday to solve the problem that affect their lives. The problem is importat that choosing a new profession if a bad decision is made then time & resources are wasted so the important that people to know how to make decision well Their is six step to to insure best decision

- **Identify the problem:-** first step towards solving problem is identify the problem Ex:- if give the Assignment or problem to student that is outside the book However, when you are doing problem solving outside the classroom you need to know to identify problem before you solve it if dont know problem you can not solve it

- **Understand the problem:-** you must understant problem that Continese toodrds solution it include knowledge based ~~of the~~ person or machine knowing knowledge base is very much importart you cannot use any instruction ouside this base. Ex tell the Address ~~to~~ of restaurant you ~~th~~

know the detail set of instruction to tell some one to find the resturant i.e know the knowledge about city you cannot solve problem calculas account you must know knowledge of Acc & cal

Identify alternative way to Solve the problem:- This list should be a complete as possible you might talk to other people to find the solution Alternative solution should be acceptable ones.

select the best way to Solve the problem from the list of altenative solution:-

In this step we identify the pros and cons of each possible solution before selecting best one for this criteria of sol. Evaluation is important.

List the instruction that enable you to solve the problem Using selected solution:-

knowledge base. instruction use to solve the proble that is known to indivisual or the machine

Evaluate the solution:- To evaluate or test the solution meas check the result to see it is correct. if it is satisfy the need of person with problem If it is unsatisfy then Review instruction again or start process all over again

people can solve the problem daily at home or work

* what to look for dinner
* which movie to see the evening
* which ~~a~~ car to buy
* How to sell the house
* work policies, management ~~customers~~

Six step

(1) identify problem How do individual spend the evening

(2) Understand the problem knowledge base participant to ~~go~~ ~~to~~ play the game and other solution

(3) identify alternative

(a) watch tv (b) invite friend (c) play vidio game (d) goto movies (e) Go to friend party

(4) Select best way to solve the problem

(a) Select alternative way that acceptable in that cost and intrest of individual are involved

(b) Specify pros & cons of remaining alter

(c) weight the pros & cons to to take final decision

(5) prepaire list of steps (for fun evening)

(6) Evaluate the solution

Are you fun yet?

if NO Review Step & process

~~if~~ must ~~stop~~ go start again

Types of problem:-

problems do not always straightforw... solved solution problem like baking cake this solution is called by series of action algorithmic solution

when we chosen best solution among several method then we completing action step these step is called algorithm.

other problem like buy the stock these solution required knowledge & experien... and process of trial & error so their is no any direct steps reach is called heuristic solution.

Problem solver use the six step for algorithmic & heuristic solution by this baking cake is easy to tell but very hard to tell the which is best stock so the problem solver need to often follow six step again & again. furthermor the same solution is correct another time so these problem solver can solve problem later Ex stock is did well in janvery poor in june

problem solving with computer:-

Solution means problem solving instruction is listed during the problem solving.

Result means outcoms of instruction

program means the set of instruction after they have been coded in perticular computer language

computer deals with algorithmic solution which is difficult and time consuming for human ~~be~~ ~~probl~~ solving complicated calculas is easy for the computer

but ① throw the ball ② How to speack English is not easy task for computer

It is solve by ~~step of~~ set of step that computer can understand.

AI (Artificial intelligence) :- This field deals with heuristic type of problem of computer this result is computer problem solving ability are similar to the human being AI expand computer field for use of ~~Computer~~ of Robotics

Until computer can be build to think like human human are work on heuristic solution and computer will process many algorithmic solution

Difficulties with problem solving :-

people have a many problems with the problem solving

* Not been taught how to solve problem
* taking wrong decision
* they not define the problem correct may not generate sufficient list of alternative
* Chasing best alternative eliminate good one
* Not use logical sequence of solution finally they evaluate incorrectly

problem solving is not easy It take practis In the long run It prove the great benifits

* when solving the problem with computer it is difficult task for the program Solver to write instruction
* task like largest number in group it is easy to tell which is largest one but many cannot explain step they follow to arrive at it
* most people say "I cant explain how I know I just know it" This explaination is not good enough for computer.

Computer specific system with the step of solution with the proper order you must assum that computer ~~cant~~ know nothing.

problem Solving aspects:-

Complete six - problem solving step in problem solving

Step1- Identify the problem

Step2:- Understand the problem

a) Understand problem

b) Description of knowledge base

Step3:- Identify alternative solution

Solution	pros	Cons
a		
b		
c		

Step4:- Select the best sol^n

Explain why did you select

step5:- list the numbered step by step instruction

1.
2.
3.
:
:
:

Step 6:- Test the solution
Does the solution work?
if not, change solution.

problem solving aspects:- Algorithm is used to manipulate the data for given problem algorithm is devided into smaller Unit- is called modules process of dividing an algorithm into modules is called modularization

Advantages

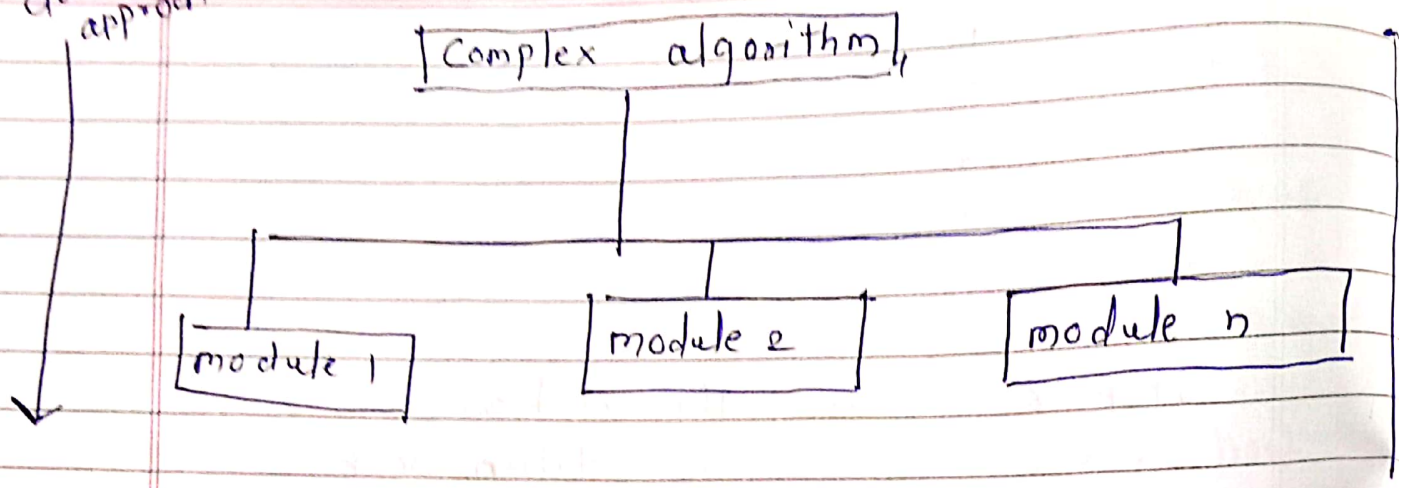
* Complex algorithm is simpler to implement
* Each module can be designed independantly which simplify implementation, debugging testing, documenting and maintenance

There is two approch to design algorithm

1. Top down approach
2. Bottom up approach

Bottom UP approach

Top down approch



Top down design:- top down design mean divide the complex algorithm into one or more module module further divide into one or more sub module It is top down refinement This design is refined into more concrete level is reached that required no further refinement

Bottom up approch:- It is revercse of the top down design we start design most basic and or concrete module And then proceed towards designing the higher level sub module is group together and form the higher level this process repeted until design is complete

problem solving statergies :-

Computer is very powerful and versatile machine but no intelligence of its own. Computer can do the task that is given by programmer wrong instruction can give error

programmer have a resposibility to give correct instruction for this he should work step by step solution to the problem step can be

1. Clearly define a problem in very simple
2. Analyze problem to find out diffr way to solve problem
3. Clearly the step in which the solution
4. write step in programming language. it can be executed by computer

design and development of correct efficient maintainable program depe on the approch is adopted

Entire program and software development process is devided into number of phases i.e SDLC software development life cycle

SDLC can be summarized

* Requirments analysis
* Design
* Implementation
* Testing
* software development training and support
* maintenance

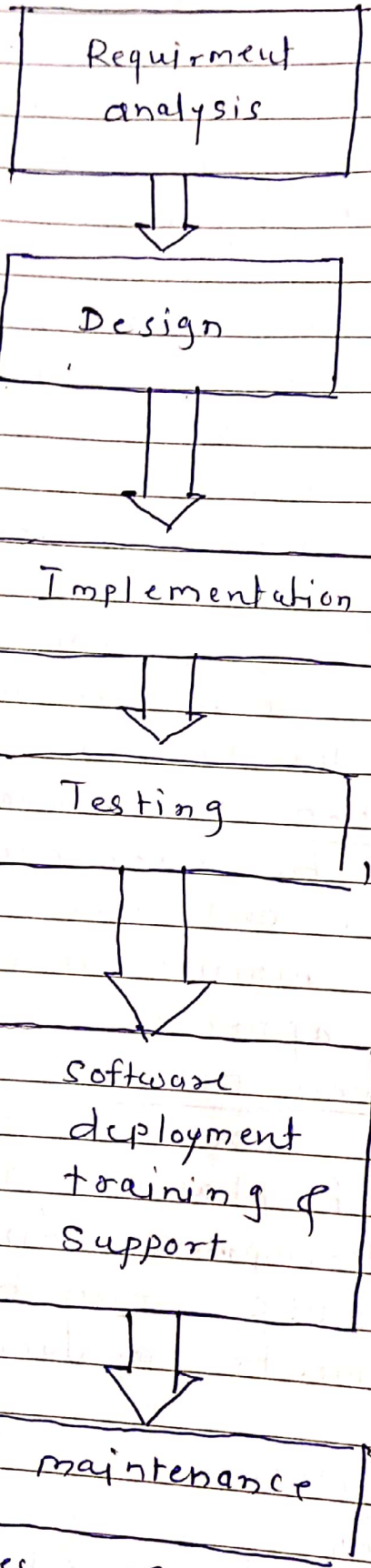


Fig:- phases of Software development lifecycle.

Requirment analysis - gather the requirment that objective of overall software product.

Design - plan of action is made before the actual development process can start the core structure of software broken down into module The solution of program for each module in the form of algorithm or flowchart

Implementation - designed algorithms are Converted into program code using any of the high-level languages. This phase called construction or code generation. while Constracting code saftware & hardware should be compatible with each other.

Testing - modules are testing together to ensure that the overall system work the software is tested using large no of input i.e. test data that ensure software is working to fullfill users requirment.

software deployment training & support - when program is approved by user then it is installed or deployed. This is crucial phase The is training is importat for this softwar because In the organization no one can know how to use it only developer ~~&pte~~ spend lots of time in developer and program designe

maintenance - maintenance & enhancement are ongoing activity because it is deals with newly discovered problem Reqired to extra piece of code for requirement to fit into original one

Basics of python programming:-

Features of python:-

① python is an exciting & powerfull language. Performance & features that make programing easy & fun.

② highleret, interpreted, interactive, object-oriented & reliable language. simple like English like

③ Vast library for complex solution.

④ It is platform independant, scripted langage

⑤ Create highpowered, high focused application

⑥ It is simple & small language. Reading a program written in python feels almost Reading English. This allows program that focus on problem rather than language itself.

⑦ Easy to learn A python is defined & easily ereadble structure of python is simple, few keyword ~~whic~~ with clearly defined Syntax so any ~~or~~ one can pick up language

⑧ Versatile means its support text process & www browsing to games

⑨ Free & open source:- Open source software freely distributed

⑩ High-level language:- when writting program in python, programmer dont have worry about low level detail i.e managing memory etc

⑪ Interactive :- programmer interact with interpreter to write a program.

⑫ portable:- This program behave the sam on variety of hardware i.e program works on different os like Linux, windows FreeBSD macintosh etc

(13) Object oriented :- it support Object oriented as well as procedure-oriented object ~~means~~ oriented data & function within object and procedure oriented means build program around procedure and function which is not reuseble.

(14) Interpreted :- simple executing & faster. It process runtime by interpreter no need to Compile You can simply run the program python convert code into intermediate bytecode which convert Computer language we just run without doing Compile, linking & loading processes.

(15) Dynamic :- executed dynamically if error its reported runtime its allow to interact with programmer, it's flexible development Enviorm

(16) Extensible :- we can add low level module to python that customize their tools to work efficiently.

(17) Embeddable :- programmer can embed Python their code within C & C++ Com, ActiveX, CORBA & java. i.e scripting cabability

(18) Extensive libraries i.e easily ~~pro~~ portable accross different platform Unix, window Macintosh perform appl^n like text processing database maintaing GUI programming.

(19) Easy maintenance :- Code written in python to maintain

(20) Secure :- It is securing enviorment from tempain

(21) Robust :- In programming errors are raised ~~exe~~ as exception can catch and handle by code Syntastical mistake can display message is displayed so the language is Roubust

(22) multitreaded :- python can executed more than one process symultaneously It also mange the task.

(22) Garbage collection :- Python having runtime Environment handle the Garbage collection i.e unuse object ~~Correctly~~ is deleted

Limitation of python
* ~~slow~~ python is slow compared c / c++
* Not for developing high graphics 3d game
* lacks of true multiprocessor support
* Difficult to big python application in single exe file

History of python
(1) python was develop by Guido van Rossum in the late 80s and early 90's at the National institute for mathematics and computer Science in the Netherland Its derived from many language ABC, modula-3, C, c++, Algol-68, smalltalk Unix shell and other scripting language
(2) In 90s python improved tremendously
(3) Version 1.0 released in 1991 & Version 2.0 in 2000 by BeOpen python lab team
(4) Version 2.7 used today support until 2020 insted of 2.8 used 2.7 further development of python 3 ~~6.4~~ currently 3.6.4 is available it is better for flexible string representation
(5) Python Copyrighted Source code available Under GNU General public license
(6) python is maintained by a core development team at institute which is directed by Guido van Rossum
(7) Python emerged as very powerfull but it is older than java, javascript

(8) Why it is called python?

It is develop February 1991 while working CWI (called stichting Mathematisch centrum)

At the time of development the comedy seres from 70's called "monty pythons Flying circus" Rossum want shoudt, slightly mysterious name. so give python.

Future Of python

(1) python is userbase & constantly growing.
(2) Stable language so its stay long.
(3) This is Preferable language for Nokia Company Nokia Google & youtube as well as Nasa for easy cynlax.
(4) OOP & parellel programming makes it ideal choice for programming
(5) python in top 5 language academics as well as indastry
(6) It is highspeed dynamic language so it is
(7) Used for photo development
(8) Company use because it has broder range of application like social n/w automation to science calculation

Writing and executing python program:-

(1) download python from ww.python.org
Python 3.4.1 & also new vergen python 3.5 & 3.6
(1) Fist Using Commandline running python interpreter directly
(2) Second Using GUI software that comes with python called python integrated Development & learning Enviornment IDLE

Print Hello world !!!

```
>>> print("Hellow world!!!")
Hellow world !!!
```

IDLE support platform windows Unix (Mac OS IDLE editor Notpad Notpad++

* Writing python programming
Step1:- Open editor
Step2:- Write the instruction
Step3:- Save file having extention .py
Step4:- Run the interpreter with Name.py
Press F5 for run then select run module

Literal Constant :-
Literal Constant we will ~~read~~ about number and string Constant in python
For ex:- 7, 3.9, 'A' & "Hellow" are literal Constant

(A) Numbers :-
You can use 4 type of number long integer, floating point & complex number
integer 5, 20 ex
long integer 5329823458999 38L
(long integer have L is suffix)
Floating point number 3.23 & 91.5E-2
Complex number a+bi
i.e -3+7i

E notation indicate power

ex - 91.5E-2 91.5 * 10-2

Comman or numeric values like
3, 5671, 23.89 -8,904, Not allowed

91.5E-2 means 91.5 * 10-2

Arithmetic overflow problem:-
2.7e200 * 4.3e200 Result is infinity

Arithmetic Underflow problem:-
3.0e-400/5.0e200 Result 0.0

0.0 is arithmetic Underflow

Loss of precision problem:-
1/3 = 0.3333... 3 inifinitely
Result is approximation for scitific problem precise calculation is required so it slitly loss of accuracy

Format function :- This fun produce string version of number with specific number of decimal place.

Without Using format

```
>>> float(16/(float(3)))
5.333333
```

Using format

```
>>> format(float(16/float(3)), '.2f')
'5.33'
```

Produe .2f Rounds result 2 decimal plac

```
>>> format(3**50, '.5e')
'7.17898e+23'
```
(for scientific Notation)

```
>>> format(123456, ',')
'123,456'
```

Format function produce numeric string of floating point value Rouded to specific number of decimal places

Simple operation on Number :-
this is directy on consol

```
>>> 10+7
26
>>> 50+40-35
55
>>> 12 * 10
120
>>> 96 / 12
8.0
>>> (-30 * 4) + 500
380
```

Division by zero :- python generate error

```
>>> 15/0
ZerodivisionError : dision by zero
```

Dividing with two integers produce Floating point number

```
>>> 5*3.0
15.0
>>> 19+3.5
22.5
```

Quotient & Remainder

division (//)

modulo operator (%)

```
>>> 78//15
15
>>> 78%5
3
>>>
```

Exponentiation :- (+, - , * & /)

support

also support ** it is used to exponentiation i.e raising one power to the another power of another

```
>>> 5**3
125
>>> 121**0.5
11.0
```

String :- String is a group of charact
if you want to use text in python
we use string

* Using single quotes (')

We can write as 'HELLO'

* Using Double quotes :- Same as the single quotes

'HELLO' & "Hello" Same

* Using Triple quotes (''' ''') :- Usin
for multiline string. In that we use
single quotes & double quotes

Ex - multiline string can be given a

''' Good morning everyone
" welcome 'python' "
Happy reading '''

We can also print by using print functio

>> 'Hello'
>> "Hello"
>> '''Hellow'''

O/P

'Hello'

String literal Concatenation
We can concatenated 3 string literal

print ('Beutiful weather''....''seems it would rain')

O/P — Beutiful weather ... Seems it would rain

Unicode string :- we can use native language like Hindi by prefixing u or U Ex.

u "Sample Unicode string"

U prefix that the file contains text written in language Other than English

+ Escape sequences :- we use character after backslash

print ('What \'s your name?')
O/P What's your name?

print ("The boy replies \"My name is Ajay.\"")

O/P The boy replies "My name is Aaditya."

print ("\") ⟹ "

print ("\\") ⟹ \

Print ("\'") ⟹ '

Ex printf(" Today is 15th August. \n India became independent on this day.")

O/p Today is 15th August.
India became independent on this day.

Ex Print(" Hello. \t welcome world of python."

O/p Hello. Welcome to the world of python

Escap sequence	Ex	O/p
\\	print("\\")	\
\'	print("\'")	'
\"	print("\"")	"
\a	print("\a")	Bell ring
\t	print("Hello \t world")	Hello world
\o	print("\056") (Print octal value)	.
\x	print("\x87") print hex value	†
\n	new line	

Raw string :- display string exactly as specified prefixing r or R

print (R "what\'s your name?")

O/P what \'s your name?

* String Formatting :- we already use build-in format() function

format (value, format_specifier)

value - string, format_specifier = formating opn

Symbols
< left justification
> right justification
^ centre justification

>>> format ('Hello', '<30')

format ('Hello', '>30)

format ('Hello', '^30')

Hello

Hello

Hello

Variable & identifier:- Variable are exa of identifier, we can store piece of info in variable it is part of memo with appropriate name.

* first character _ (undrscore) or lett

* rest of identifier can be ('_'), uppercase, lowercase letter digit (0-9

* Also case sensitive myvar and my are not same

* @, $ and % not allowed with identifier

Valid identifier
Sum, _My_Var, num1, r, Var_20
First etc

invalid identifier
1num, My-var, % check, Basic Sa
H#R$A etc

Data types :- we have see the variable that can be hold values of different types called datatype

For example person age store number
Address is mixtur numbers & characters

Basic type is numbre & string
standard data type numbers, string, list, tuple dictionary

(A) Initializing value to variable
Assigning value to veriable by = sign

Left side is name
Right side is value store

```
val = 'Hello'
print (val)
Val = 100
print (val)
val = 12.34
print (val)
```

O/p

Hello
100
val

```
>> x = 5
>> y = 10
>> print ('Hello')
>> print (x+y)
```

Hello
15

(B) multiple Assignment
Assign single value multiple variab

sum = flag = a = b = 0
(All integer variable assign 0)

you can also asign different value to multiple variable

Sum, a, b, mesg = 0, 3, 5, "Result"

```
Sum = 0
a = 3
b = 5
mesg = "RESULT"
```

different value to multiple variable

Removing variable by del

```
Str = "Hello"
age = 10
num = 20
print (str)
  Hello
print (num)
20
print (age)
20
```

age not declard

```
Str = "Hello"
num = 10
age = 20
print (Str) → Hello
print (num) → 10
print (age) → 20
del num
Print (num)
Error
num is not defined
```

(C) multiple statements on single line
if you want specify more than one statement in sigle line then you should use a semicolon to separet two statements

(D) Boolean :- Boolean is data type in the python having two values True or False

ex >>> Boolean_var = True
>>> print (Boolean_var)
True

ex >>> 20 == 30
false

ex >>> 90 <= 90
True

ex "python" == "python"
True

ex >>> 87 >= 87.0
False

ex >>> 20 != 20
False

* **Input Operation**:- In the real word we need program interactive :

To take input from Users, Python use input() function input function takes Users input as a string. So whether you input a number or String, it is treated as string only.

```
name = input("What's is your name?")
age = input("Enter your age:")
print(name + ", you are" + age + "years old")
```

O/P

```
What's your name? Ganesh
Enter your age: 20
```

⇒ Ganesh, you are 10 years old

* **Comments**:- Comment is non executable statement in program. It is discription of the code. Comment make program readable and understandable for programmer and User.

(#) Sign begin the Comment. All character following # upto end of line are part of command.

```
# This is comment
print ("Hello") # to display hello
# program ends here
```

O/P

⇒ Hello

* **Reserved word**:- In programming language there is predefined meaning this word is called Reserved words or keywords [Not use Identifier]

Reserved word ⇒ and, assert, break, class, Continue, def, del, elif, else, except, from, if import, in, is, lambda, Print, raise, return, try while with [keyword is Lowercase letter only]

* Indentation :- Whitespace at the beginning of the line is called indentation use single tab for each indentation level of that logical line

```
age = 21
    print (" You can Vote")
    ^
```

Unexpected indent

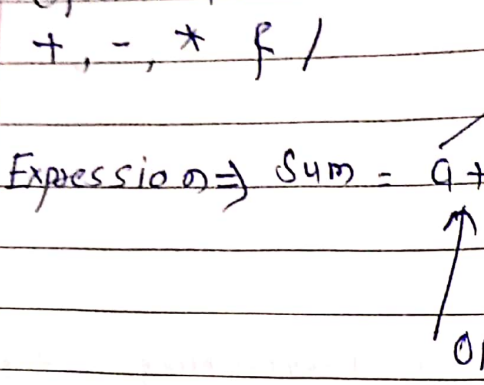
Statement in the block have same indentation level python very strictly checks the indentation level and give an error if indentation is not correct.

All the statement inside block same indentation level

[^ indicate error symbols]

* **Operators and expression:-**

operators are manipulate the value of operands. Some basic operator include +, -, * & /



(A) Arithmetic Operators

Arithmetic operators are +, -, *, /, %, * *, and // a = 100, b = 200

Operator	Discription	Ex	O/P
+	Addition	print(a+b)	300
-	Substraction	print(a-b)	-100
*	multiplication	print(a*b)	20000
/	Division	print(b/a)	2.0
%	modulo operator return remainder	Print(b % a)	0
//	Divide Return Quotient if Operand is negative then result floored	print(12//5)	2
		print(12.0//5.0)	2.0
		print(-19//5)	-4
		print(-20.0//3)	-7.0

Exponent print(a**b) 100^{200}

* *

(B) Comparison operators use to compare the operands its also called relational operators

a = 100 b = 200

Operator	Example	O/p
==	print(a==b)	false
!=	print(a!=b)	True
>	print(a>b)	False
<	print(a<b)	true
>=	print(a>=b)	false
<=	print(a<=b)	True

* Shortcut Operator

c = a

a+=b is same as a=a+b

a-=b is same as a=a-b

a*=b — " — a=a*b

a/=b — " — a=a/b

a%=b — " — a=a%b

a//=b — " — a=a//b

a**=b — " — a=a**b

* Unary operators python support Unary minus operator it is act on the single operand

```
b = 20
a = -(b)        Result    a = -20
                Now value is -20
```

* Bitwise Operators
bitwise AND, bitwise OR, bitwise XOR, Shift opera

```
10101010 & 01010101 = 00000000 (AND)
10101010 | 01010101 = 11111111 (OR)
10101010 ^ 01010101 = 11111111 (XOR)
~10101010 = 01010100 (Bitwise Not)
```

A	B	A&B	A/B	A^B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

A	/A
0	1
1	0

shift operators - python support shift left << & shift right (>>)

if we have x = 0001 1101 then
X << 1 gives result = 0011 1010

* Logical operators
Python support logical operators Logical AND (&&) logical (OR) (||) and logical Not (!)

~~(a > b) && (b > c)~~

Logical AND (&&) :- if both the expression is true then whole expression true

Logical OR (||) :- if one or both expression of the logical operator is true then whole expression is true

Logical Not (!) logical not operator negat the value of expression.

a = 10, b

b = !a

(a > b) && (b > c) both should be true

(a > b) || (b > c) A one should be true

Membership Operators :- python support 2 types of membership Operators in & not in It test membership in sequence such as String list

in Operator :- It return true if variable found in sequence Otherwise false

Ex a in nums return 1 if member of nums

not in Operator :- The operator return true if variable not found

Ex a not in nums return 1 if a is not member of nums

identity Operator :- This Operators compare the memory locations of 2 object

Is Operator :- return 1 if id(a) is same id(b)

Is not Operator :- if a is not b return 1 if id(a) is not same as id(b)

Operators precedence & Associativity

** Exponentiation
~ complement + unary plus - minus
* / //

BEDMAS
↳ bracket, exponetion, division, multi
Addition & subst n

>> 10 + 30 * 5

* has higher precedence than +

70 * 3

>>> (40 + 20) * 30 / 10
180 ~~+0/10~~ 60 * 3
180

>>> ((40 + 20) * 30) / 10
180 1800 / 10

>>> (40 + 20) * (30 / 10)
60 * 3
180

>>> 40 + (20 * 30) / 10
~~600~~
40 + 600/10
40 + 60
100

Operator	Description
**	Exponentiation
~ + -	Complement Unary plus Unary minus
*, /, %, //	mutiply, divide, modulo & floar division
+, -	Addition & substraction

python performs operations in the same order as that of normal mathematics

BEDMAS

B → bracket
E → exponentiation
D → division
M → multiplication
A → Addition
S → Substraction

Expression in python :

Expression must have at least one Operand and one or more operators.

Operand is value on which operator is applied

Ex — A * B + C - 5

*, +, - Operators

A B C Variable

$x = a / b$ $y = a * b$ $z = a \wedge b$

$x = a > b$

Types of Expression

Based on the position of operator in expression

* infix Expression :- In most commanly used type Operator placed in beth operands

$a = b - c$

* prefix Expression :- operators is placed befor operands $a = \square bc$

* postfix Expression :- Operator Placed After Operand $a = bc\square$

Based on data type of the result obtained on evaluating an expression

Constant expression → $8 + 9 - 2$ (Invalue constant)

Integral Expression → $a = 10$ $b = 5$ $c = a * b$ (Produce integer expres

Floating point expression → produce floating point resul

$a * b / 2$

Relation Expression → return true false

$c = a > b$

Logical Expression :→ two more relational expres and return a value true & false

$a > b \;\&\&\; y! = 0$

Assignment Expression :- Assign the variable

$c = a + b$ or $c = 20$