Chapter 5
Python – Variables and Operators

PART – 1
I. Choose The Best Answer

Question 1.
Who developed Python?
(a) Ritche
(b) Guido Van Rossum
(c) Bill Gates
(d) Sunder Pitchai
Answer:
(b) Guido Van Rossum

Question 2.
The Python prompt indicates that Interpreter is ready to accept instruction?
(a) >>>
(b) <<<
(c) #
(d) <<
Answer:
(a) >>>

Question 3.
Which of the following shortcut is used to create new Python Program?
(a) Ctrl + C
(b) Ctrl + F
(c) Ctrl + B
(d) Ctrl + N
Answer:
(d) Ctrl + N

Question 4.
Which of the following character is used to give comments in Python Program?
(a) #
(b) &
(c) @
(d) $
Answer:
(a) #

Question 5.
This symbol is used to print more than one item on a single line?
(a) Semicolon
(b) Dollor($)
(c) Comma(,)
(d) Colon(;)
Answer:
(c) Comma(,)

Question 6.
Which of the following is not a token?
(a) Interpreter
(b) Identifiers
(c) Keyword
(d) Operators
Answer:
(a) Interpreter

Question 7.
Which of the following is not a Keyword in Python?
(a) Break
(b) While
(c) Continue
(d) Operators
Answer:
(d) Operators

Question 8.
Which operator is also called as Comparative operator?
(a) Arithmetic
(b) Relational
(c) Logical
(d) Assignment
Answer:
(b) Relational

Question 9.
Which of the following is not Logical operator?
(a) And
(b) Or
(c) Not
(d) Assignment
Answer:
(d) Assignment

Question 10.
Which operator is also called as Conditional operator?
(a) Ternary
(b) Relational
(c) Logical
(d) Assignment
Answer:
(a) Ternary

PART – II
II. Answer The Following Questions

Question 1.
What are the different modes that can be used to test Python Program?
Answer:
In Python, programs can be written in two ways namely Interactive mode and Script mode. The Interactive mode allows us to write codes in Python command prompt (>>>) whereas in script mode programs can be written and stored as separate file with the extension .py and executed. Script mode is used to create and edit python source file.

Question 2.
Write short notes on Tokens?
Answer:
Python breaks each logical line into a sequence of elementary lexical components known as Tokens. The normal token types are;

1. Identifiers
2. Keywords
3. Operators
4. Delimiters
5. Literals

Whitespace separation is necessary between tokens, identifiers or keywords.

Question 3.
What are the different operators that can be used in Python?
Answer:
In computer programming languages operators are special symbols which represent computations, conditional matching etc. The value of an operator used is called operands. Operators are categorized as Arithmetic, Relational, Logical, Assignment etc.

Question 4.
What is a literal? Explain the types of literals?
Answer:

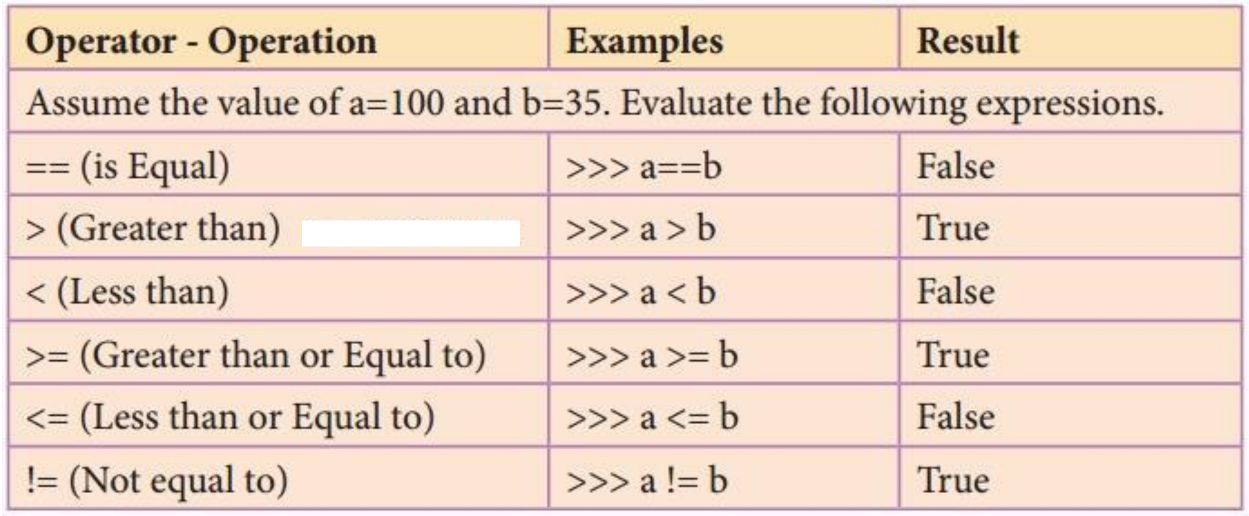
Literal is a raw data given in a variable or constant. In Python, there are various types of literals.

1. Numeric
2. String
3. Boolean

Question 5.
Write short notes on Exponent data?
Answer:
An Exponent data contains decimal digit part, decimal point, exponent part followed by one or more digits.
12.E04, 24.e04 # Exponent data

PART – III
III. Answer The Following Questions

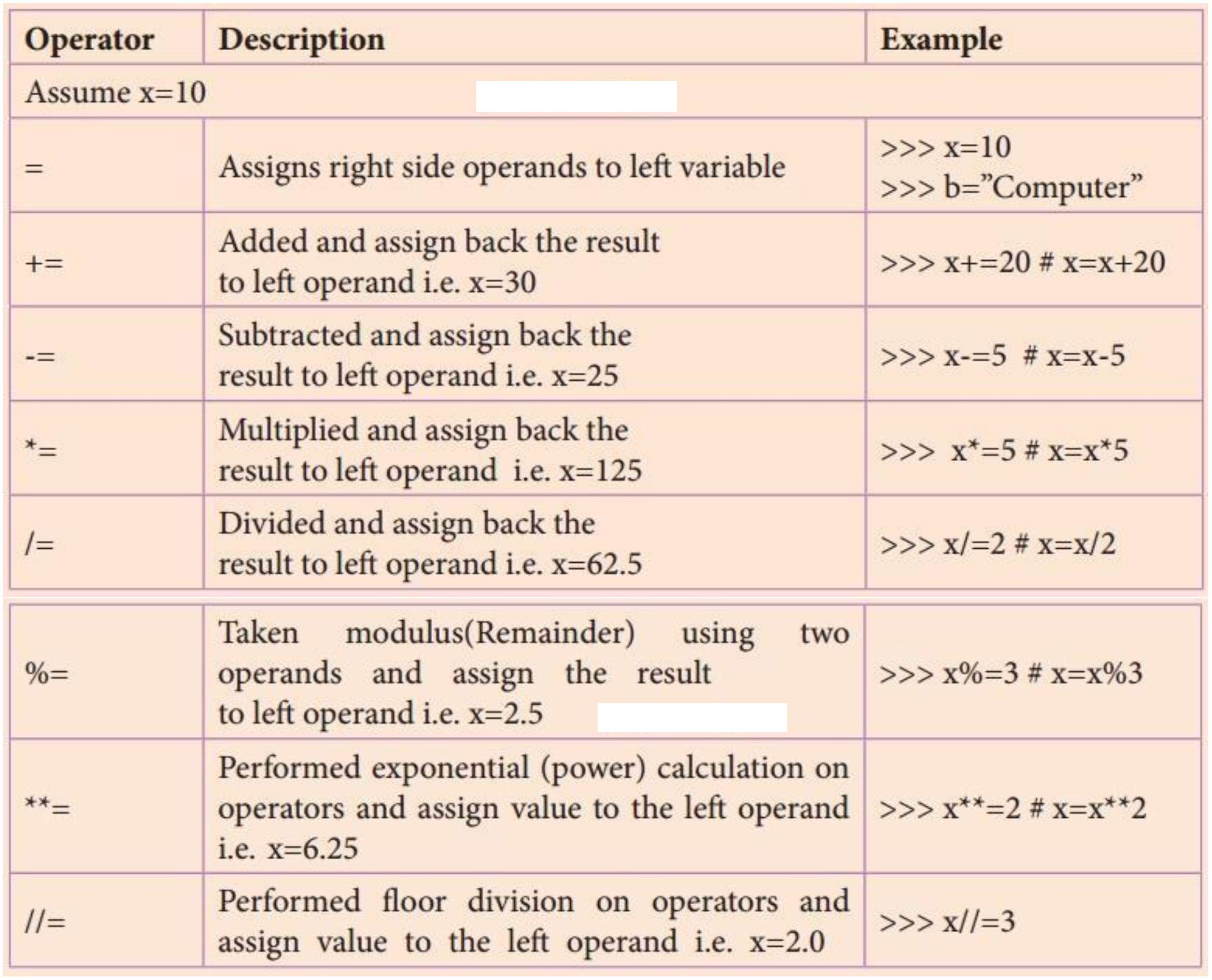
Question 1.
Write short notes on Arithmetic operator with examples?
Answer:
Arithmetic operators:
An arithmetic operator is a mathematical operator that takes two operands and performs a calculation on them. They are used for simple arithmetic. Most computer languages contain a set of such operators that can be used within equations to perform different types of sequential calculations.
Python supports the following Arithmetic operators.



Operator - Operation	Examples	Result
Assume the value of a=100 and b=35. Evaluate the following expressions.		
== (is Equal)	>>> a==b	False
> (Greater than)	>>> a > b	True
< (Less than)	>>> a < b	False
>= (Greater than or Equal to)	>>> a >= b	True
<= (Less than or Equal to)	>>> a <= b	False
!= (Not equal to)	>>> a != b	True

Question 2.
What are the assignment operators that can be used in Python?
Answer:

Assignment operators:
In Python, = is a simple assignment operator to assign values to variable. Let a = 5 and b = 10 assigns the value 5 to a and 10 to b these two assignment statement can also be given as a, b = 5, 10 that assigns the value 5 and 10 on the right to the variables a and b respectively. There are various compound operators in Python like + =, – =,* =, / = % =,** = and //= are also available.



Operator	Description	Example
Assume x=10		
=	Assigns right side operands to left variable	>>> x=10 >>> b="Computer"
+=	Added and assign back the result to left operand i.e. x=30	>>> x+=20 # x=x+20
-=	Subtracted and assign back the result to left operand i.e. x=25	>>> x-=5 # x=x-5
*=	Multiplied and assign back the result to left operand i.e. x=125	>>> x*=5 # x=x*5
/=	Divided and assign back the result to left operand i.e. x=62.5	>>> x/=2 # x=x/2
%=	Taken modulus(Remainder) using two operands and assign the result to left operand i.e. x=2.5	>>> x%=3 # x=x%3
**=	Performed exponential (power) calculation on operators and assign value to the left operand i.e. x=6.25	>>> x**=2 # x=x**2
//=	Performed floor division on operators and assign value to the left operand i.e. x=2.0	>>> x//=3

Question 3.
Explain Ternary operator with examples?
Answer:
Conditional operator:
Ternary operator is also known as conditional operator that evaluate something based on a condition being true or false. It simply allows testing a condition in a single line replacing the multiline if – else making the code compact.
The Syntax conditional operator is,
Variable Name = [on – true] if [Test expression] else [on – false]
Example:
min = 50 if 49 < 50 else 70 # min = 50 min = 50 if 49 > 50 else 70 # min = 70

Question 4.
Write short notes on Escape sequences with examples?
Answer:
Escape Sequences:
In Python strings, the backslash “\” is a special character, also called the “escape” character. It is used in representing certain whitespace characters: “\t” is a tab, “\n” is a newline, and “\r” is a carriage return. For example to print the message “It’s raining”, the Python command is >>> print (“It\’s rainning”)
It’s rainning
Python supports the following escape sequence characters.

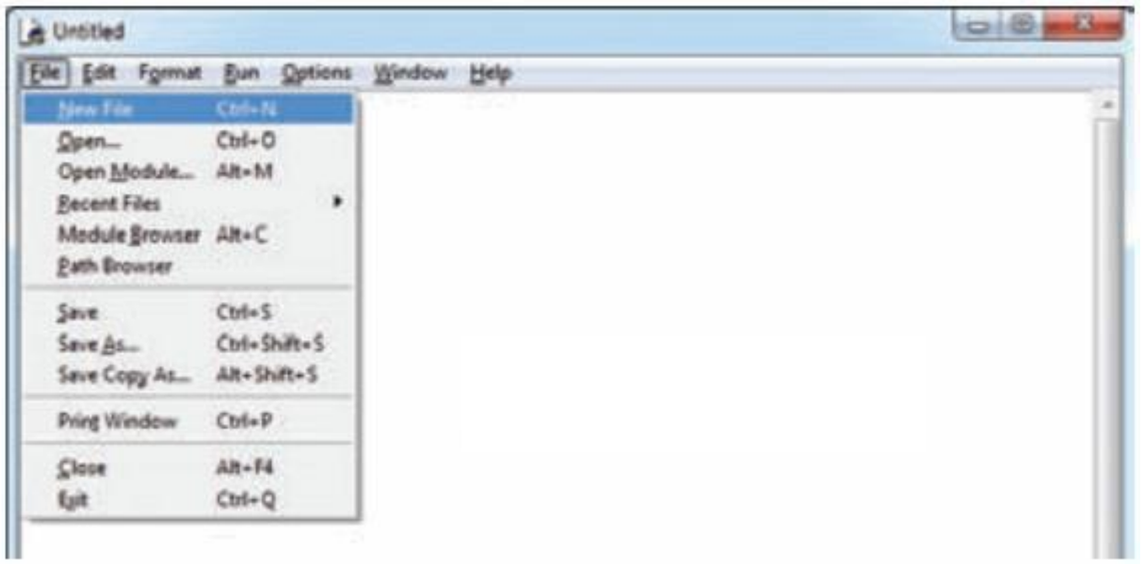
Escape sequence character	**Description**	**Example**	**Output**
\\	Backslash	>>> print("\\test")	\test
\’	Single-quote	>>> print("Doesn\'t")	Doesn't
\”	Double-quote	>>> print("\"Python\"")	"Python"
\n	New line	print("Python","\n","Lang..")	Python Lang..
\t	Tab	print("Python","\t","Lang..")	Python Lang..

Question 5.
What are string literals? Explain?
Answer:
String Literals:
In Python a string literal is a sequence of characters surrounded by quotes. Python supports single, double and triple quotes for a string. A character literal is a single character surrounded by single or double quotes. The value with triple – quote is used to give multi – line string literal.
Strings = “This is Python”
char = “C”
multiline _ str = “This is a multiline string with more than one line code”.

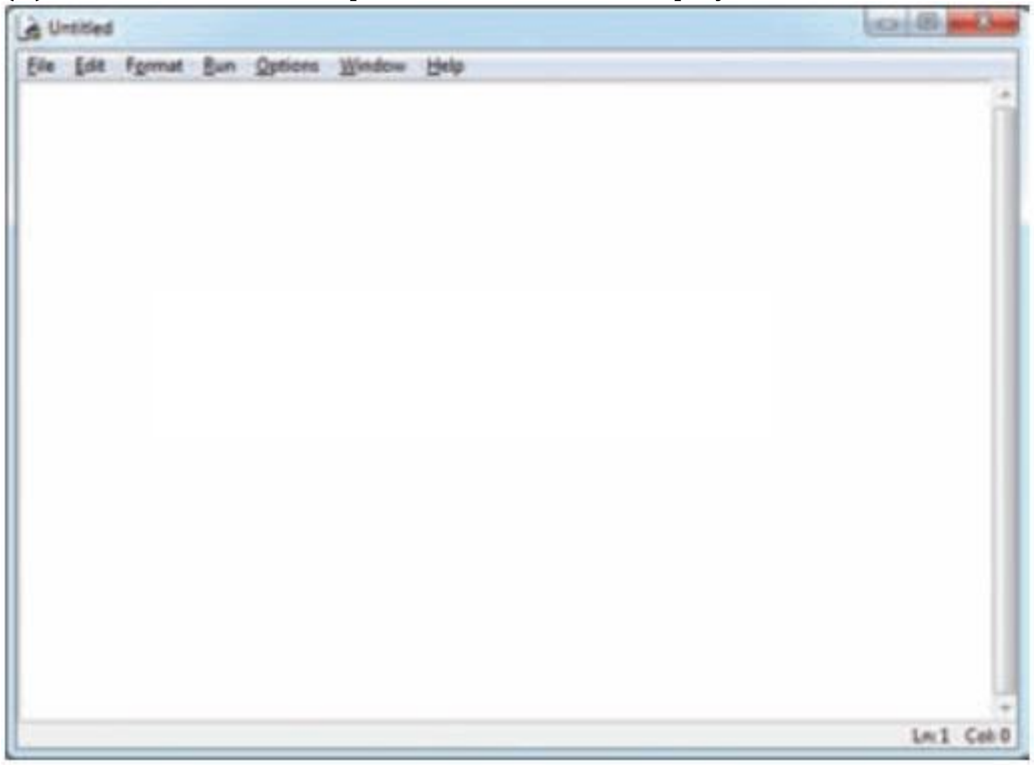
PART – IV
IV. Answer The Following Questions

Question 1.
Describe in detail the procedure Script mode programming?
Answer:
Script mode Programming:
Basically, a script is a text file containing the Python statements. Python Scripts are

reusable code. Once the script is created, it can be executed again and again without retyping. The Scripts are editable.
Creating Scripts in Python:
(I) Choose File → New File or press Ctrl + N in Python shell window.

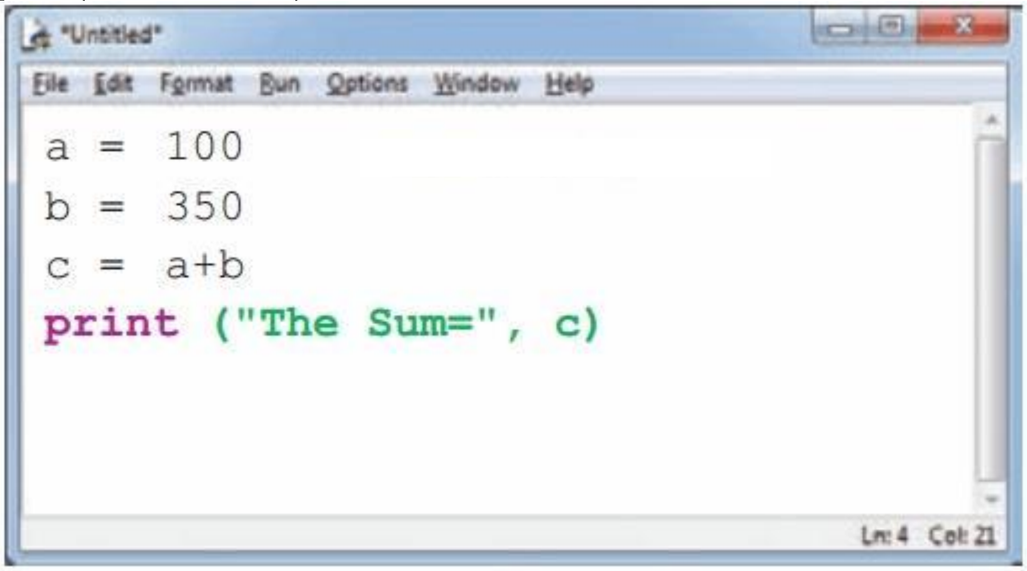


(II) An untitled blank script text editor will be displayed on screen



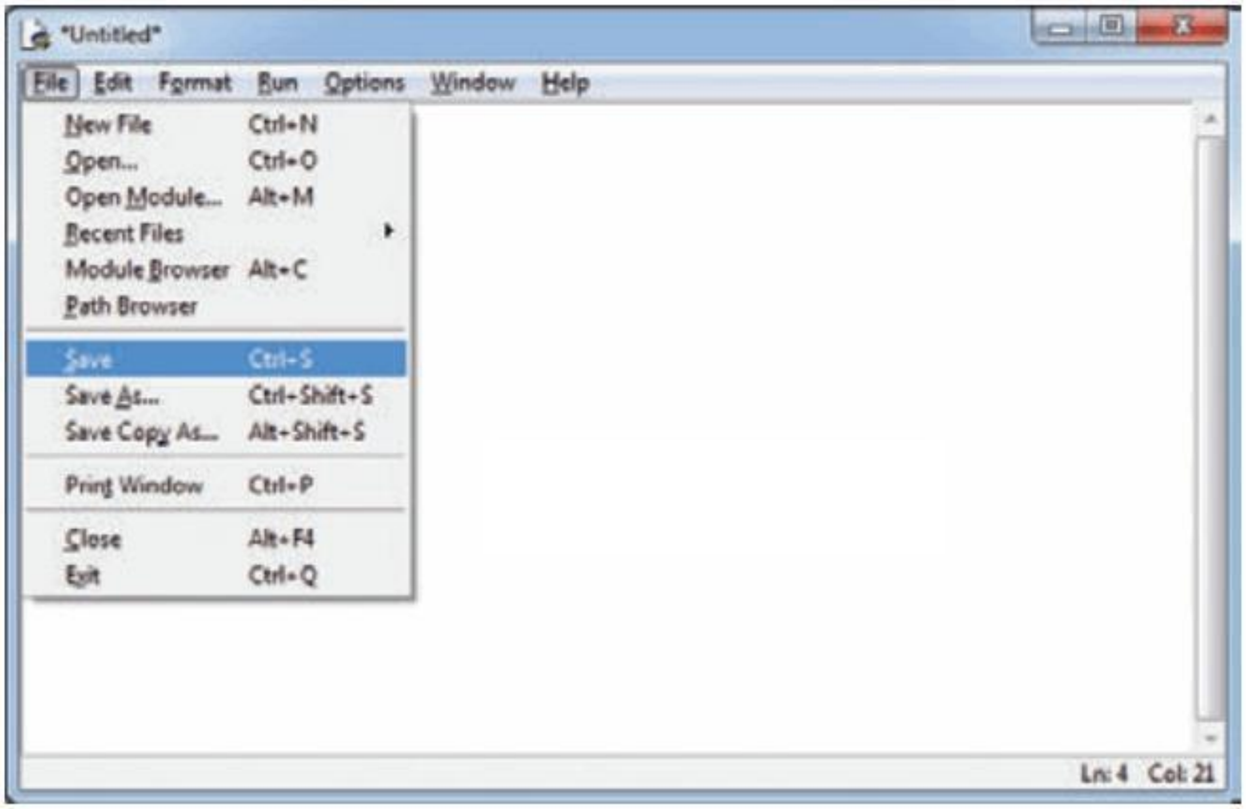
(III) Type the following code in Script editor
a = 100

b = 350
c = a + b
print (“The Sum = “, c)

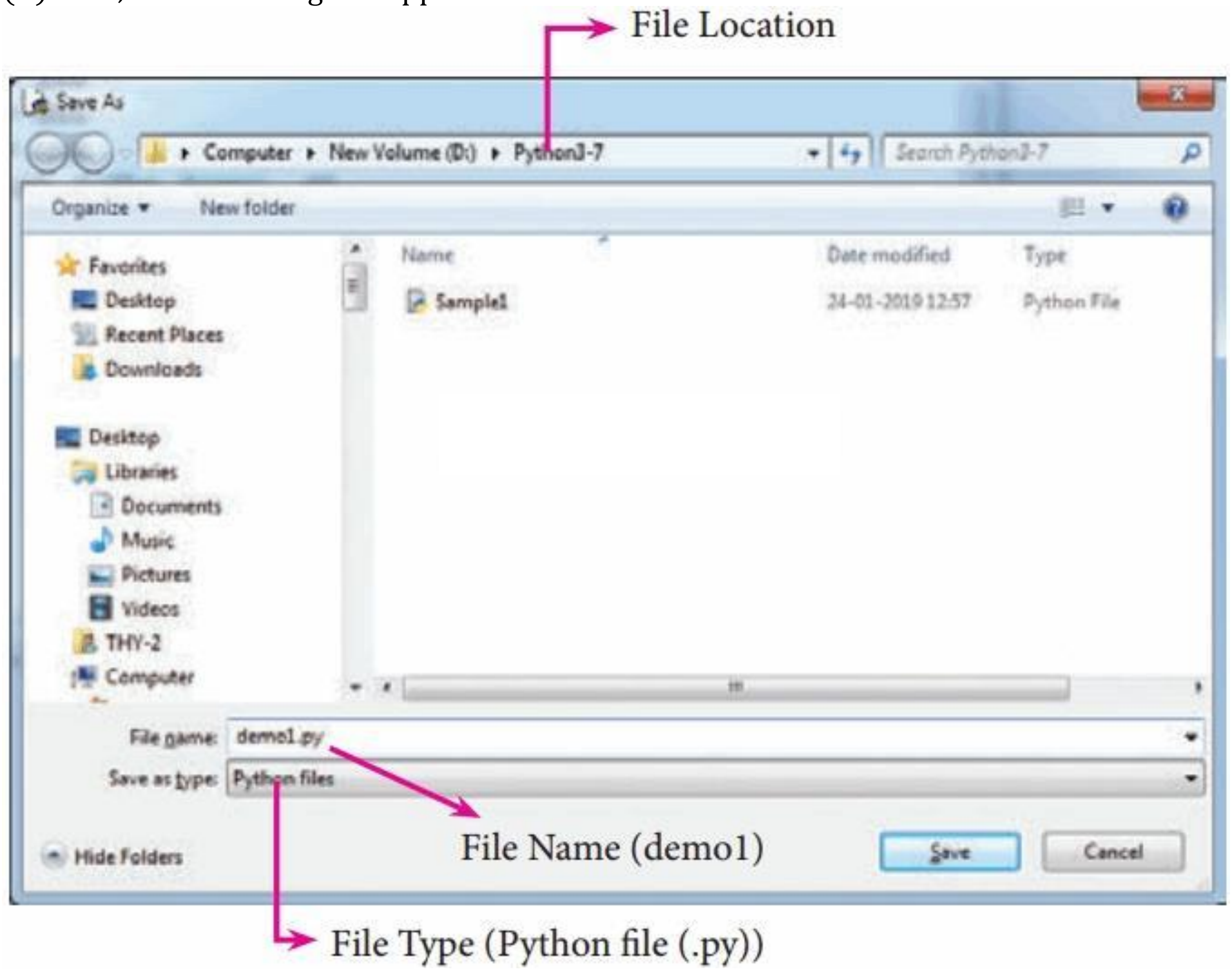


```
a = 100
b = 350
c = a+b
print ("The Sum=", c)
```

Saving Python Script
(I) Choose File → Save or Press Ctrl + S



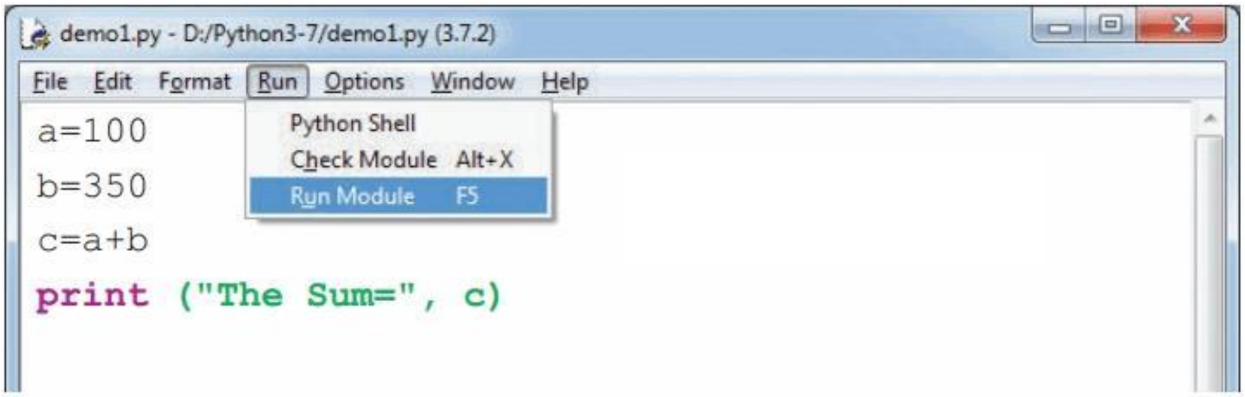
(II) Now, Save As dialog box appears on the screen



(III) In the Save As dialog box, select the location where you want to save your Python code, and type the file name in File Name box. Python files are by default saved with extension .py. Thus, while creating Python scripts using Python Script editor, no need to specify the file extension.

(IV) Finally, click Save button to save your Python script.
Executing Python Script

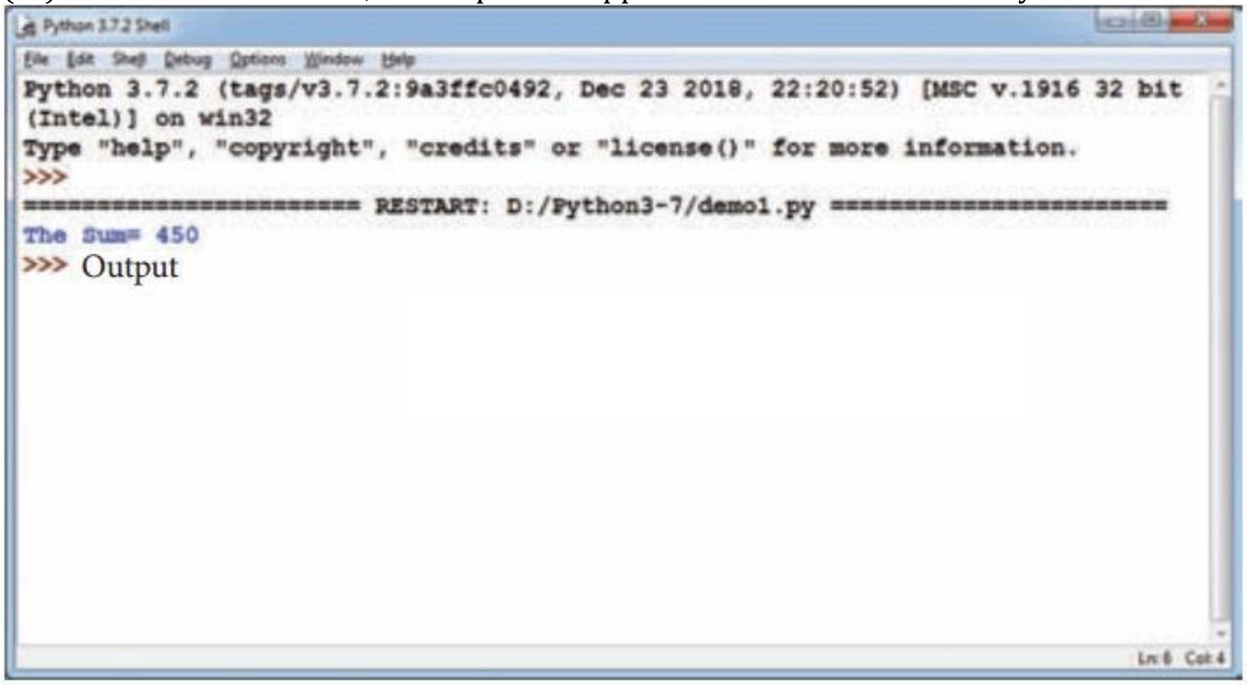
(I) Choose Run → Run Module or Press F5


demo1.py - D:/Python3-7/demo1.py (3.7.2)
File Edit Format Run Options Window Help
Python Shell
Check Module Alt+X
Run Module F5

```
a=100
b=350
c=a+b
print ("The Sum=", c)
```

(II) If your code has any error, it will be shown in red color in the IDLE window, and Python describes the type of error occurred. To correct the errors, go back to Script editor, make corrections, save the file using Ctrl + S or File → Save and execute it again.

(III) For all error free code, the output will appear in the IDLE window of Python:



```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======================= RESTART: D:/Python3-7/demo1.py =======================
The Sum= 450
>>> Output
```

Question 2.
Explain input ( ) and print ( ) functions with examples?
Answer:
Input and Output Functions:
A program needs to interact with the user to accomplish the desired task; this can be achieved using Input – Output functions. The input ( ) function helps to enter data at run time by the user and the output function print ( ) is used to display the result of the program on the screen after execution.
The print ( ) function

In Python, the print Q function is used to display result on the screen. The syntax for print Q is as follows:
Example
print (“string to be displayed as output ”)
print (variable)
print (“String to be displayed as output ”, variable)
print (“String1 ”, variable, “String 2”, variable, “String 3”)
Example
```
>>> print (“Welcome to Python Programming”)
Welcome to Python Programming
>>> x = 5
>>> y = 6
>>> z = x + y
>>> print (z)
11
>>> print (“The sum = ”, z)
The sum = 11
>>> print (“The sum of”, x, “and”, y, “is”, z)
The sum of 5 and 6 is 11
```
Th print ( ) evaluates the expression before printing it on the monitor. The print ( ) displays an entire statement which is specified within print ( ). Comma (,) is used as a separator in print ( ) to print more than one item.
input ( ) function
In Python, input ( ) function is used to accept data as input at run time. The syntax for input ( ) function is,
Variable = input (“prompt string”)
Where, prompt string in the syntax is a statement or message to the user, to know what input can be given.
If a prompt string is used, it is displayed on the monitor; the user can provide expected data from the input device. The input ( ) takes whatever is typed from the keyboard and stores the entered data in the given variable. If prompt string is not given in input ( ) no message is displayed on the screen, thus, the user will not know what is to be typed as input.

Example 1:
input ( ) with prompt string
```
>>> city = input (“Enter Your City: ”)
Enter Your City: Madurai
>>> print (“I am from “, city)
I am from Madurai
```

Example 2:
input ( ) without prompt string
```
>>> city = input ( )
Rajarajan
>>> print (I am from”, city)
```

```
I am from Rajarajan
```
Note that in example – 2, the input ( ) is not having any prompt string, thus the user will not know what is to be typed as input. If the user inputs irrelevant data as given in the above example, then the output will be unexpected. So, to make your program more interactive, provide prompt string with input ( ).
The input ( ) accepts all data as string or characters but not as numbers. If a numerical value is entered, the input values should be explicitly converted into numeric data type. The int ( ) function is used to convert string data as integer data explicitly. We will leam about more such functions in later chapters.

Example 3:
```
x = int (input("Enter Number 1: "))
y = int (input("Enter Number 2: "))
print ("The sum =", x + y)
```
Output:
```
Enter Number 1:34
Enter Number 2:56
The sum = 90
```

Example 4:
Alternate method for the above program
```
x, y = int (input("Enter Number 1 :")), int(input("Enter Number 2:"))
print ("X = ",x,"Y = ",y)
```
Output:
```
Enter Number 1:30
Enter Number 2:50
X = 30 Y= 50
```

Question 3.
Discuss in detail about Tokens in Python?
Answer:
Tokens:
Python breaks each logical line into a sequence of elementary lexical components known as Tokens. The normal token types are

1. Identifiers
2. Keywords
3. Operators
4. Delimiters
5. Literals.

Whitespace separation is necessary between tokens, identifiers or keywords.

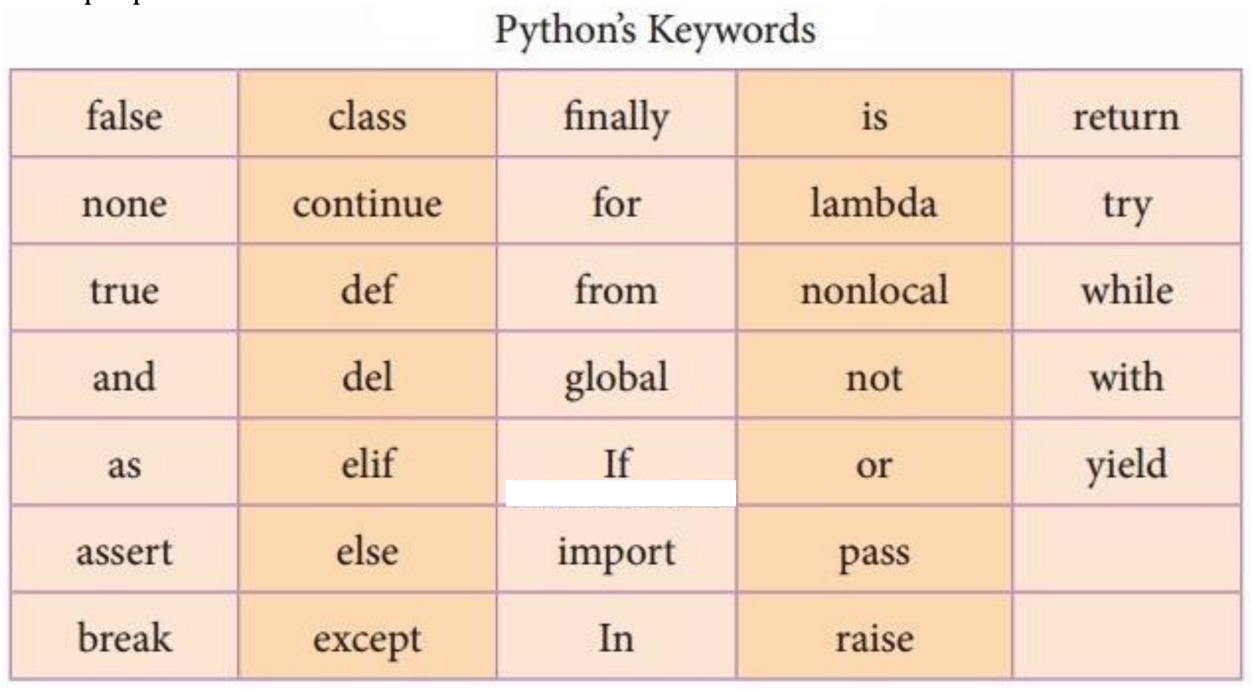
(I) Identifiers:
An Identifier is a name used to identify a variable, function, class, module or object.

1. An identifier must start with an alphabet (A..Z or a..z) or underscore (_).
2. Identifiers may contain digits (0 .. 9)
3. Python identifiers are case sensitive i.e. uppercase and lowercase letters are distinct.
4. Identifiers must not be a python keyword.
5. Python does not allow punctuation character such as %, $, @ etc., within identifiers.

Example of valid identifiers
Sum, total _ marks, regno, num 1

Example of invalid identifiers
12 Name, name$, total – mark, continue

(II) Keywords
Keywords are special words used by Python interpreter to recognize the structure of program. As these words have specific meaning for interpreter, they cannot be used for any other purpose.

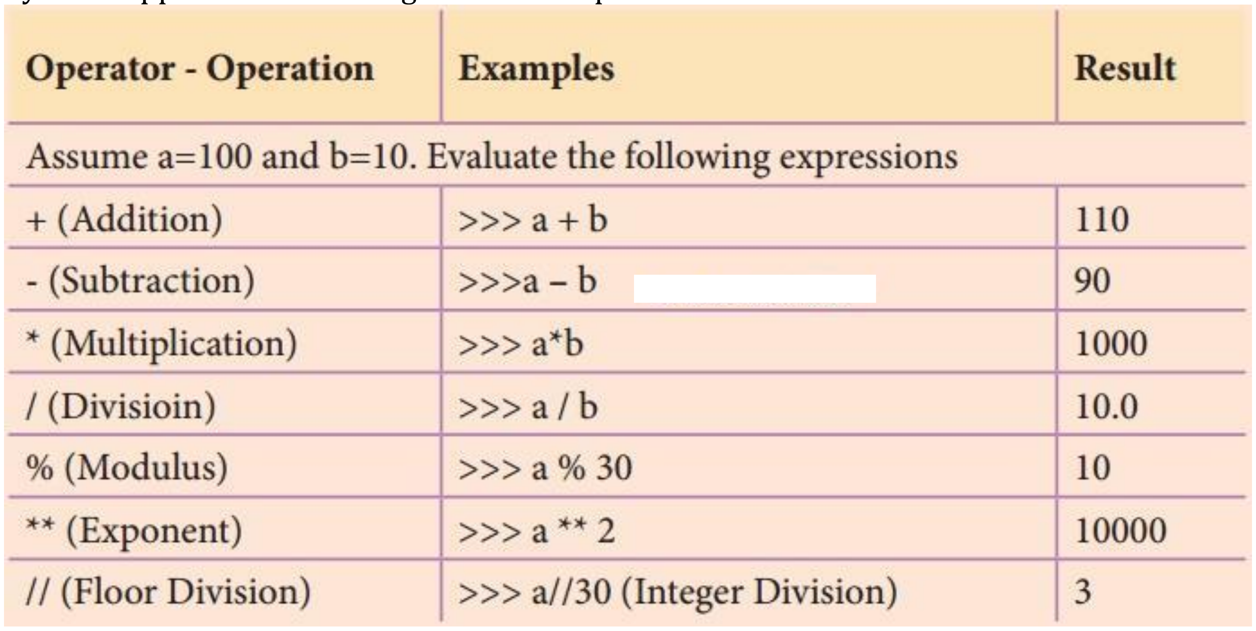

Python's Keywords

false	class	finally	is	return
none	continue	for	lambda	try
true	def	from	nonlocal	while
and	del	global	not	with
as	elif	If	or	yield
assert	else	import	pass	
break	except	In	raise	

(III) Operators
In computer programming languages operators are special symbols which represent computations, conditional matching etc. The value of an operator used is called operands. Operators are categorized as Arithmetic, Relational, Logical, Assignment etc. Value and variables when used with operator are known as operands.

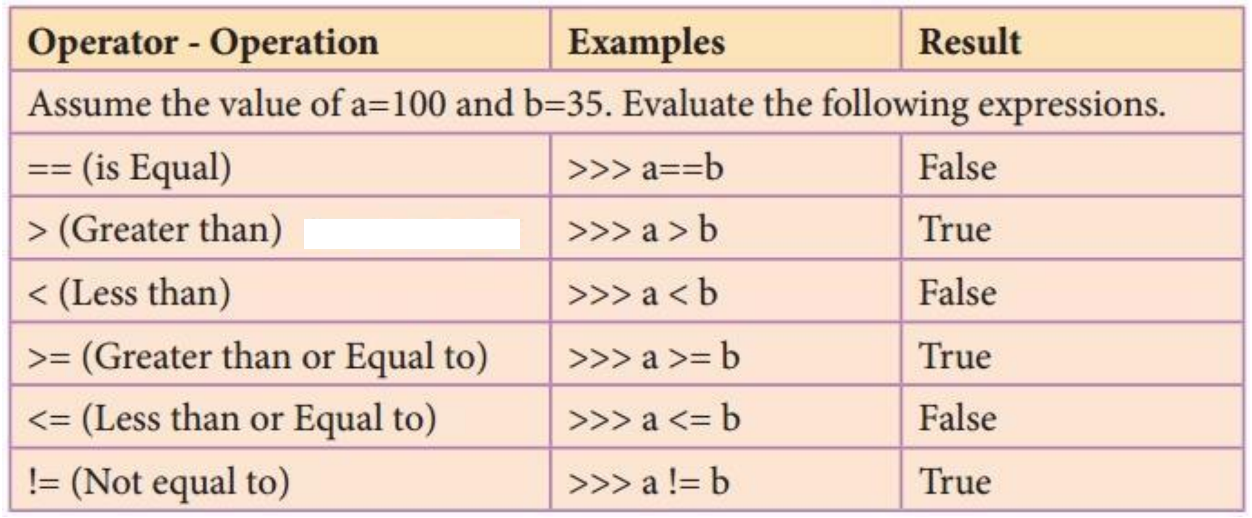
Arithmetic operators:
An arithmetic operator is a mathematical operator that takes two operands and performs a

calculation on them. They are used for simple arithmetic. Most computer languages contain a set of such operators that can be used within equations to perform different types of sequential calculations.
Python supports the following Arithmetic operators.



Operator - Operation	Examples	Result
Assume a=100 and b=10. Evaluate the following expressions		
+ (Addition)	>>> a + b	110
- (Subtraction)	>>>a – b	90
* (Multiplication)	>>> a*b	1000
/ (Divisioin)	>>> a / b	10.0
% (Modulus)	>>> a % 30	10
** (Exponent)	>>> a ** 2	10000
// (Floor Division)	>>> a//30 (Integer Division)	3

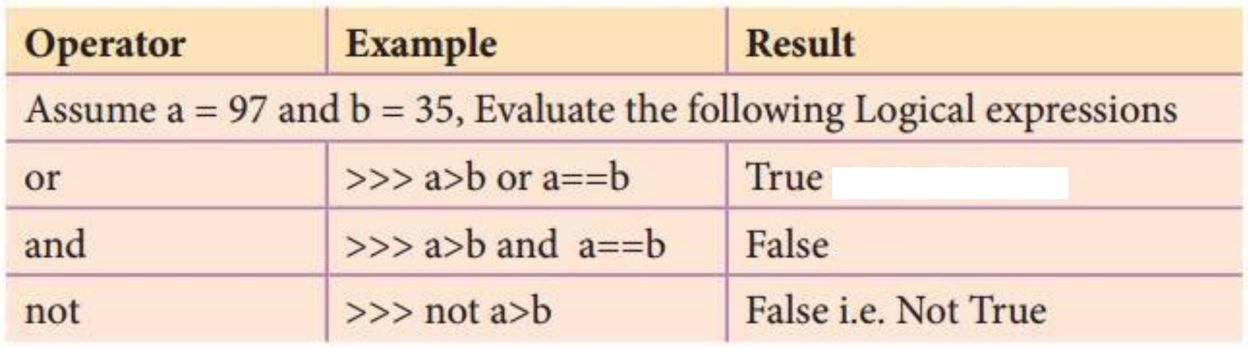
Relational or Comparative operators:
A Relational operator is also called as Comparative operator which checks the relationship between two operands. If the relation is true, it returns True; otherwise it returns False.



Operator - Operation	Examples	Result
Assume the value of a=100 and b=35. Evaluate the following expressions.		
== (is Equal)	>>> a==b	False
> (Greater than)	>>> a > b	True
< (Less than)	>>> a < b	False
>= (Greater than or Equal to)	>>> a >= b	True
<= (Less than or Equal to)	>>> a <= b	False
!= (Not equal to)	>>> a != b	True

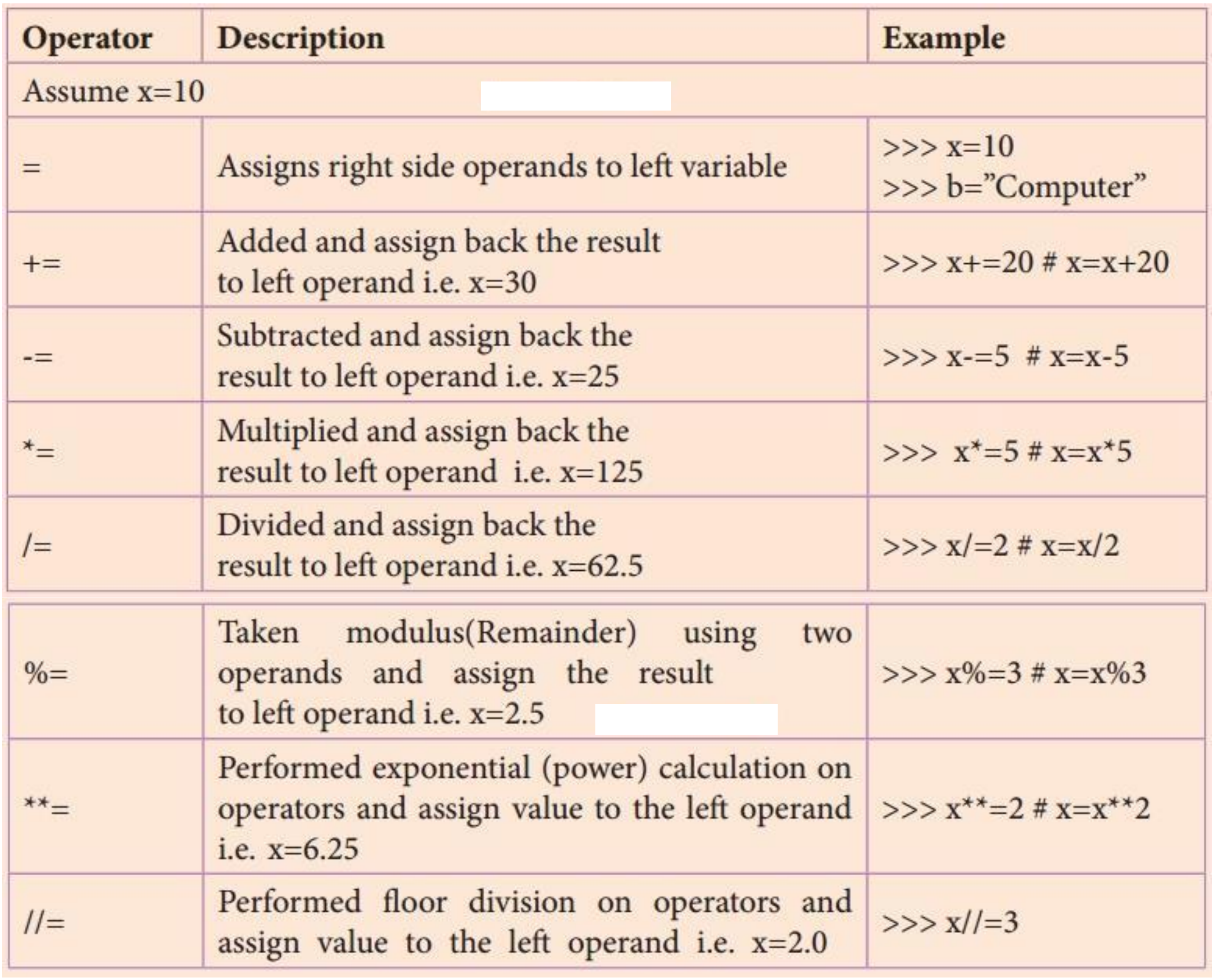
Logical operators:
In python, Logical operators are used to perform logical operations on the given relational

expressions. There are three logical operators they are and, or and not.



Operator	Example	Result
Assume a = 97 and b = 35, Evaluate the following Logical expressions		
or	>>> a>b or a==b	True
and	>>> a>b and a==b	False
not	>>> not a>b	False i.e. Not True

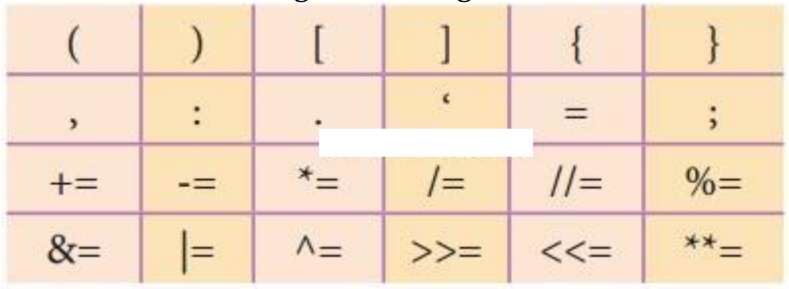
Assignment operators:
In Python, = is a simple assignment operator to assign values to variable. Let a = 5 and b = 10 assigns the value 5 to a and 10 to b these two assignment statement can also be given as a, b = 5, 10 that assigns the value 5 and 10 on the right to the variables a and b respectively. There are various compound operators in Python like + =, – =, * =, / =, % =, ** = and //= are also available.



Operator	Description	Example
Assume x=10		
=	Assigns right side operands to left variable	>>> x=10 >>> b="Computer"
+=	Added and assign back the result to left operand i.e. x=30	>>> x+=20 # x=x+20
-=	Subtracted and assign back the result to left operand i.e. x=25	>>> x-=5 # x=x-5
*=	Multiplied and assign back the result to left operand i.e. x=125	>>> x*=5 # x=x*5
/=	Divided and assign back the result to left operand i.e. x=62.5	>>> x/=2 # x=x/2
%=	Taken modulus(Remainder) using two operands and assign the result to left operand i.e. x=2.5	>>> x%=3 # x=x%3
**=	Performed exponential (power) calculation on operators and assign value to the left operand i.e. x=6.25	>>> x**=2 # x=x**2
//=	Performed floor division on operators and assign value to the left operand i.e. x=2.0	>>> x//=3

Conditional operator:
Ternary operator is also known as conditional operator that evaluate something based on a condition being true or false. It simply allows testing a condition in a single line replacing the multiline if – else making the code compact.
The Syntax conditional operator is,
Variable Name = [on _ true] if [Test expression] else [on _ false]
Example:
min = 50 if 49 < 50 else 70 # min = 50 min = 50 if 49 > 50 else 70 # min = 70

(IV) Delimiters
Python uses the symbols and symbol combinations as delimiters in expressions, lists, dictionaries and strings. Following are the delimiters.



(	)	[	]	{	}
,	:	.	'	=	;
+=	-=	*=	/=	//=	%=
&=	|=	^=	>>=	<<=	**=

(V) Literals
Literal is a raw data given in a variable or constant. In Python, there are various types of literals.

1. Numeric
2. String
3. Boolean

1. Numeric Literals:
Numeric Literals consists of digits and are immutable (unchangeable). Numeric literals can belong to 3 different numerical types Integer, Float and Complex.

2. String Literals:
In Python a string literal is a sequence of characters surrounded by quotes. Python supports single, double and triple quotes for a string. A character literal is a single character surrounded by single or double quotes. The value with triple-quote is used to give multi-line string literal.

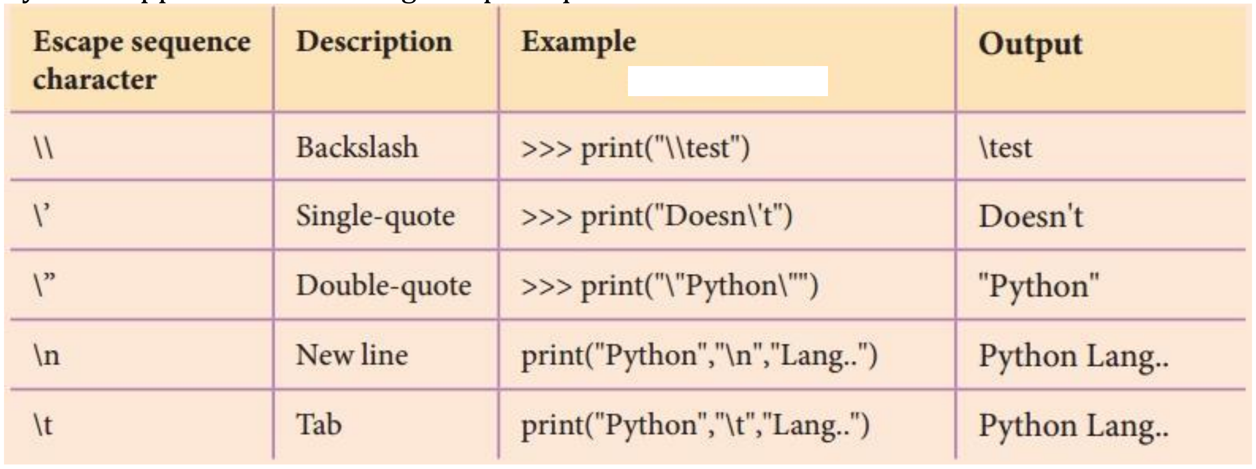
3. Boolean Literals:
A Boolean literal can have any of the two values: True or False.

Escape Sequences:
In Python strings, the backslash "\" is a special character, also called the "escape" character. It is used in representing certain whitespace characters: "\t" is a tab, "\n" is a newline, and

“\r” is a carriage return. For example to print the message “It’s raining”, the Python command is

```
>>>print (“It\’s rainning”)
It’s rainning
```

Python supports the following escape sequence characters.



Escape sequence character	Description	Example	Output
\\	Backslash	>>> print("\\test")	\test
\’	Single-quote	>>> print("Doesn\'t")	Doesn't
\”	Double-quote	>>> print("\"Python\"")	"Python"
\n	New line	print("Python","\n","Lang..")	Python Lang..
\t	Tab	print("Python","\t","Lang..")	Python Lang..