ASSIGNMENT-2
CHAPTER5:PYTHON–VARIABLESANDOPERATORS PART I

Multiple Choice Questions:

1. Who developed Python?
a. Ritche **b. Guido Van Rossum** c. Bill Gates d. Sundar Pitchai
2. The Python Prompt that indicates the interpreter is ready to accept instruction is …………….
a. >>> b. <<< c. # d. <<
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**
4. Which of the following shortcut is used to save Python program?
a. **Ctrl+S** b. Ctrl+V c. Ctrl+C d.Ctrl+N
5. Python files are saved with extension …………
a. **.py** b. .pr c. .pq d. .ps
6. This symbol used to print more than one item on a single line is ………..
a. Semicolon(;) b. Dollar($) c**. Comma(,)** d. Colon(:)
7. Which of the following Character is used to give comments in Python program?
a. **#** b. & c. @ d. $
8. Which operator is also called as comparative operator?
a. Arithmetic b. **Relational** c. Logical d. Assignment
9. Which Operator is also called as Conditional Operator?
a. **Ternary** b. Relational c. Logical d. Assignment
10. Which function is used to display result on the screen?
a. **print( )** b.input ( ) c. sum ( ) d. mult ()

PART II

Very Short Answers:

1. What are the different modes that can be used to test Python program?

In Python, programs can be written in two ways namely
Interactive mode &
Script mode.
In interactive mode allows us to write codes in Python command prompt **(>>>)**
In script mode programs can be written and stored as separate file with the extension **.py** and executed

2. What is a literal? What are the types of literals?

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

a) Numeric Literals: It consists of digits and are immutable (unchangeable).
b) String literal: It is a sequence of characters surrounded by quotes.
c) A Boolean literal: It can have any of the two values: True or False.

3. Write short notes on 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 and

5) Literals.

Whitespace separation is necessary between tokens, identifiers or keywords.

4. What are the different operators that can be used in Python?

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**.

5. Write short notes on Exponent data?

A floating point data is represented by a sequence of decimal digits that includes a decimal point. An Exponent data contains decimal digit part, decimal point, exponent part followed by one or more digits.

Example :

123.34, 456.23, 156.23	# Floating point data
12.E04, 24.e04	*# Exponent data*

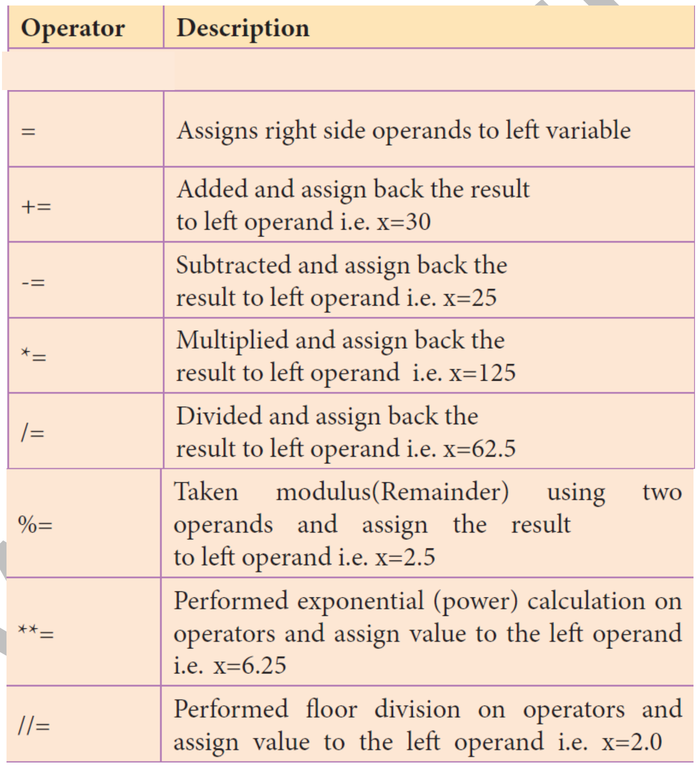
PART III

Short Answers:

1. What are the assignment Operators that can be used in Python?

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.

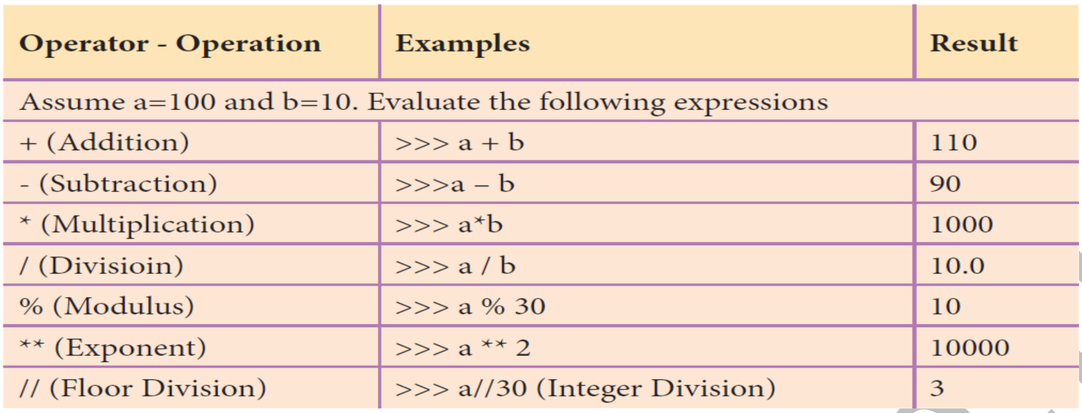
There are various compound operators in Python like +=, -=, *=, /=, %=, **= and //= are also available.



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

2. Write short note on Arithmetic operators with examples.

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

3. Write short notes on Escape Sequence with example?

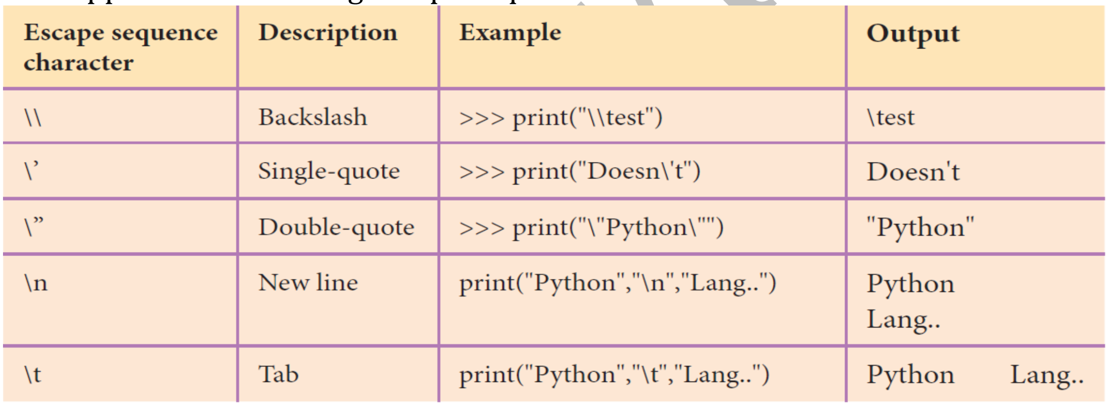
In Python strings, the backslash **"\\"** is a special character, also called the **"escape"** character. It is used in representing certain whitespace characters: **"\t"** is a tab, **"\n"** is a newline, and **"\r"** is a carriage return. For example to print the message "It's raining", the

Python command is

>>> print ("It\'s rainning")

It's rainning

Python supports the following escape sequence characters.

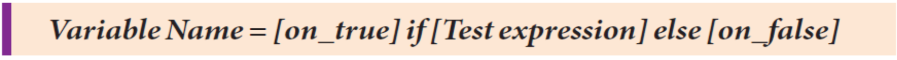


Escape sequence character	Description	Example	Output
\\	Backslash	>>> print("\\test")	\test
\’	Single-quote	>>> print("Doesn\'t")	Doesn't
\”	Double-quote	>>> print("\"Python\"")	"Python"
\n	New line	print("Python","\n","Lang..")	Python Lang..
\t	Tab	print("Python","\t","Lang..")	Python Lang..

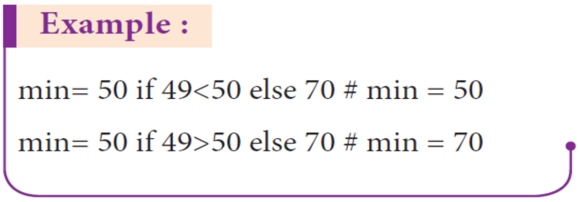
4. Explain Ternary Operators with examples.

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
```

5. What are String Literals? Explain.

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 multiline string literal.

```
# Demo Program to test String Literals
        strings = "This is Python"
        char = "C"
        multiline_str = "'This is a multiline string with more than one line code.'"
        print (strings)
        print (char)
        print (multiline_str)
# End of the Program
```

PART IV

Long Answers:

1. Discuss in detail about Tokens in Python?

Python breaks each logical line into a sequence of elementary lexical components known as **Tokens.** The normal token types are
Identifiers, Keywords, Operators, Delimiters and Literals.
Whitespace separation is necessary between tokens, identifiers or keywords.

Identifiers

An Identifier is a name used to identify a variable, function, class, module or object.
An identifier must start with an alphabet (A..Z or a..z) or underscore (_).

- Identifiers may contain digits (0 .. 9)
- Python identifiers are case sensitive i.e. uppercase and lowercase letters are distinct.
- Identifiers must not be a **python** keyword.
- Python does not allow punctuation character such as %,$, @ etc., within identifiers.

Example of valid identifiers

Sum, total_marks, regno, num1

Example of invalid identifiers

12Name, name$, total-mark, continue

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.

Example

False class finally is return none continue for
lambda try True def from non local
while

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

i) Arithmetic Operators
ii) Relational Operators
iii) Logical Opeators
iv) Assignment Operators

Value and variables when used with operator are known as **operands.**

1) 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. Python supports the following Arithmetic operators

- + (Addition),
- (Subtraction),
- (Multiplication),
- / (Divisioin),
- % (Modulus),
- ** (Exponent),
- // (Floor Division)

2) Relational 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**. Python supports following relational operators

- == (is Equal),
- (Greater than),
- < (Less than),
- >= (Greater than or Equal to),
- <= (Less than or Equal to),
- != (Not equal to)

3) Logical Operators:

Logical operators are used to perform logical operations on the given relational expressions.

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

4) Assignment Operators:

= 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.

2. Explain input() and print() functions with examples.

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()** function is used to display result on the screen. The syntax for **print()** is as follows:

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” ……)

The **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.
Example
>>> city=input ("Enter Your City: ")
Enter Your City: Madurai
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.
x = int (input("Enter Number 1: "))

Prepared By
S.Gowri Raamalakshmi M.C.A., B.Ed
grlakshmi30@gmail.com

CHAPTER 6: CONTROL STRUCTURES
PART I

Multiple Choice Questions:

a) How many important control structures are there in python?
a) **3** b) 4 c) 5 d)6

2) Which statement is the simplest of all decision making statements?
a) if - else b) if..elseif **c) simple if** d) Nested if

3) Which punctuation should be used in the blank? if<condition>____
statement-block 1 else:
statement-block 2
a) ; **b):** c) :: d) !

4) elif can be considered to be an abbreviation of …………
a) Elseif b) if c) simple if d) None

5) What plays a vital role in python programming?
a) Statements b) control c) structure **d) indentation**

6) How many types of loops are there in python?
a) 2 b) 3 c) 4 d) 5

7) Which is the entry check loop?
a) For b) while **c) a & b** d) None

8) The condition in the if statement should be in the form of ………….
a) Arithmetic or Relational expression b) Arithmetic or Logical expression
c) Relational or Logical expression d) Arithmetic

9) Which is the most comfortable loop?
a) do-while b) while **c) for** d) if…elif

10) A loop placed within another loop is called as……………………….?
a) for loop b) while loop **c) Nested loop** d) do-while loop

PART II

Very Short Answers:

1. Define Control Structures?

A program statement that causes a jump of control from one part of the program to another is called control structure or control statement. As you have already learnt in C++, these control statements are compound statements used to alter the control flow of the process or program depending on the state of the process.

2. Write the syntax of if -else statement?

if <condition>:
statements-block 1
else:
statements-block 2

3. Write note on range() in loop?

range() generates a list of values starting from **start** till **stop-1.**

The syntax of range() is as follows:

range (start,stop,[step])

Where,

start – refers to the initial value

stop – refers to the final value

step – refers to increment value, this is optional part.

4. Name the 3 jump statements in Python?

There are three keywords to achieve jump statements in Python : **break, continue, pass.**

5. Write a note on break statement?

The **break** statement terminates the loop containing it. The control of the program flows to the statement immediately after the body of the loop. If break statement is inside a nested loop (loop inside another loop), break will terminate the innermost loop.

Syntax:

```
    break
```

PART III

Short Answers:

1. Write a note on if -else structure.

if .. else statement

Answer:

The if., else statement provides control to check the true block as well as the false block. It thus provides two possibilities and the condition determines which BLOCK is to be executed.

Following is the syntax of 'if.else' statement.

Syntax:

```
if:
        statements – block 1
else:
            statements – block 2
```

2. Write the syntax of while loop.

Syntax:

```
while <condition>:
        statements block 1
[else:
        statements block2]
```

3. What is the output of the following code?

```
for i in range (2,10,2):
   print (i, end =' ')
```

Output:

```
2 4 6 8
```

4. Write the differences between branching and looping?

Branching Statement	Looping Statement
The statements that help us to skip a segment or set of statements and execute another segment based on the test of a condition is called **branching statements**	The statements that help us to execute set of statements repeatedly are called as **looping statements.**
They are also called as ***alternative*** Statements	They are also called as ***iterative*** Statements
Python provides the three types of	Python provides two types of looping

alternative or branching statements: (1) Simple if statement (2) if..else statement (3) if..elif statement	constructs: (1) while loop (2) for loop

5. List the differences between break and continue statement.

The **break** statement terminates the loop containing it. Control of the program flows to the statement immediately after the body of the loop.	Continue statement is used to skip the remaining part of a loop and start with next iteration.
Syntax : break	Syntax : continue

PART IV

Long Answers:

1. Write a detail note on for loop.

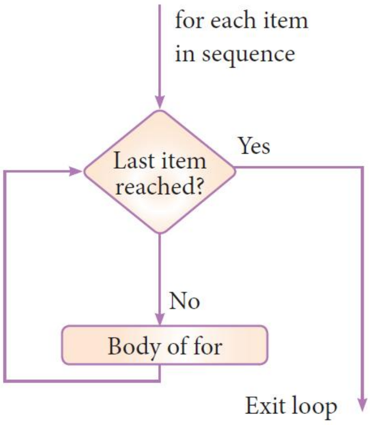
for loop:

for loop is the most comfortable loop. It is also an entry check loop. The condition is checked in the beginning and the body of the loop(statements – block 1) is executed if it is only True otherwise the loop is not executed.

Syntax:

```
    for counter _ variable in sequence:
        statements block 1
     # optional block
    [else:
       statements block 2]
```

The counter_variable mentioned in the syntax is similar to the control variable that we used in the for loop of C++ and the sequence refers to the initial, final and increment value.



Usually in Python, **for** loop uses the *range()* function in the sequence to specify the initial, final and increment values. *range()* generates a list of values starting from **start** till **stop-1.**

The syntax of range O is as follows:

range (start, stop, [step])

Where,

start – refers to the initial value

stop – refers to the final value
step – refers to increment value, this is optional part.

```
# program to illustrate the use of for loop – to print single digit even number
    for i in range (2, 10, 2):
        print (i, end = ' ')
```

Output:

```
2 4 6 8
```

2. Write a python program to display all 3 digit odd numbers.

Odd Number (3 digits)

```
for a in range (100, 1000)
    if a % 2 = = 1:
        print b
```

Output:

101, 103, 105, 107, .. …… 997, 999

Prepared By
S.Gowri Raamalakshmi M.C.A., B.Ed
grlakshmi30@gmail.com

CHAPTER 7: PYTHON FUNCTIONS PART I

Multiple choice questions

1. A named blocks of code that are designed to do one specific job is called as ……………
   a) Loop b) Branching **c) Function** d) Block
2. Which function is called as anonymous unnamed function ……………………………….?
   a) **Lambda** b) Recursion c) Function d) Define
3. A function which calls itself is called as ……………
   a) Built-in **b) Recursion** c) Lambda d) Return
4. Which of the following keyword is used to begin the function block ?
   a) define b) for c) finally **d) def**
5. While defining a function which of the following symbol is used?
   a) ; b) . **c) :** d) $
6. Which of the following keyword is used to exit a function block?
   a) define **b) return** c) finally d) def

7.…………………. function returns an absolute value of a number?
   a) **abs( )** b) ord( ) c) chr( ) d) bin( )

8. Python……………………….should not be used as function name ?
   a) **key words** b) arguments c) variables d) scope
9. A variable with …………………………..can be used anywhere in the program?
   a) local scope **b) global scope** c) class scoped) function scope
10. Any number of return statements are allowed in a function definition but ……… of them is executed at run time?
    a) **only one** b) two c) three d) four

PART II

Very Short Answers:

1. What is function?

Functions are named blocks of code that are designed to do specific job. If you need to perform that task multiple times throughout your program, you just call the function dedicated to handling that task.

2. Write the different types of function?

1. User – defined Functions
2. Built – in Functions
3. Lambda Functions
4. Recursion Functions

3. What are the advantages of function?

Main advantages of functions are:

1. It avoids repetition and makes high degree of code reusing.
2. It provides better modularity for your application.

4. Write short notes about sqrt( ) function with example?

Function: sqrt( ).
Description: sqrt ( ) Returns the square root of x. where x must be greater than 0 .
Syntax : sqrt (x)
Example:

```
      import math
      a= 30
  print (math.sqrt (a))
```

Output:

5.477225575051661

5. Write about return statement.

The return statement causes your function to exit and returns a value to its caller. Only one return statement is executed at run time even though the function contains multiple return statements.

Syntax of return: return [expression list]

This statement can contain expression which gets evaluated and the value is returned. If there is no expression in the statement or the return statement itself is not present inside a function, then the function will return the None object.

PART III

Short Answers:

1. Write short notes about anonymous functions?

Anonymous function is a function that is defined without a name. While normal functions are defined using the **def** keyword, in Python anonymous functions are defined using the **lambda** keyword. Hence, anonymous functions are also called as **lambda** functions.

Syntax of Anonymous Functions

lambda [argument(s)] :expression

Uses of lambda or anonymous function

- Lambda function is mostly used for creating small and one-time anonymous function.
- Lambda functions are mainly used in combination with the functions like filter(), map() and reduce().

2. Write the rules of local variables?

Rules of local variable:

1. A variable with local scope can be accessed only within the function/block that it is created in.
2. When a variable is created inside the function/block, the variable becomes local to it.
3. A local variable only exists while the function is executing.
4. The formal arguments are also local to function.

3. Write the basic rules for global keyword in python?

The basic rules for global keyword in Python are:

1. When we define a variable outside a function, it’s global by default. You don’t have to use global keyword.
2. We use global keyword to read and write a global variable inside a function.
3. Use of global keyword outside a function has no effect

4. Differentiate floor( ) and ceil( ) function?

Floor ()	Ceil ()
Returns the largest integer less than or equal to x	Returns the smallest integer greater than or equal to x
Syntax : math.floor (x)	**Syntax** : math.ceil (x)

Example:
import math
x=26.7
print (math.floor (x))
	Example:
	import math
	x= 26.7
	print (math.ceil (x))

Output: 26	**Output:** 27

5. How recursive function works?

- Recursive function is called by some external code.
- If the base condition is met then the program gives meaningful output and exits.
- Otherwise, function does some required processing and then calls itself to continue recursion.

PART IV

Long Answers:

1. Explain the following built in functions. id ( ), chr ( ), round ( ), type ( ), pow ( )

Function	Description	Syntax	Example
id ( )	id( ) Return the "identity" of an object. i.e. the address of the object in memory.	id (object)	x=15 print ('address of x is :',id (x)) **Output:** address of x is : 1357486752
chr ( )	chr ( ) Returns the Unicode character for the given ASCII value. This function is inverse of ord() function.	chr (i)	c=65 d=43 print (chr (c)) prin t(chr (d)) **Output:** A +
round ( )	Returns the nearest integer to its input. 1. First argument (number) is used to specify the value to be rounded. 2. Second argument (ndigits) is used to specify the number of decimal digits desired after rounding.	round (number [,ndigits])	x= 17.9 print ('x value is rounded to', round (x)) **Output:1** x value is rounded to 18 print (round (n1,0)) **Output:2** 18.0
type( )	Returns the type of object for the given single object. **Note:** This function used with single object parameter.	type (object)	x= 15.2 print (type (x)) **Output:** <class 'float'>
pow ( )	Returns the computation of ab i.e. (a**b) a raised to the power of b.	pow (a,b)	a= 5 print (pow (a,b)) **Output:** 25

2. Explain recursive function with an example.

Python recursive functions

When a function calls itself is known as recursion. Recursion works like loop but sometimes it makes more sense to use recursion than loop. You can convert any loop to recursion.

A recursive function calls itself. Imagine a process would iterate indefinitely if not stopped by some condition! Such a process is known as infinite iteration. The condition that is applied in any recursive function is known as base condition. A base condition is must in every recursive function otherwise it will continue to execute like an infinite loop.

Working Principle:

1. Recursive function is called by some external code.
2. If the base condition is met then the program gives meaningful output and exits.
3. Otherwise, function does some required processing and then calls itself to continue recursion. Here is an example of recursive function used to calculate factorial.

Example:

```
def fact (n):
if n = = 0:
  return 1
else:
  return n * fact (n – 1)
print (fact (0))
print (fact (5))
```

Output:

```
1
120
```

Prepared By
S.Gowri Raamalakshmi M.C.A., B.Ed
grlakshmi30@gmail.com

CHAPTER– 8:STRINGSANDSTRINGSMANIPULATION PART I

Choose the best Answer:

1. Which of the following operator is used for concatenation?
   a) + b) & c) * d) =
2. Defining strings with in triple quotes allows creating ………….
   a) Single line strings **b) Multiple line strings** c) Double line strings d) Multiple strings
3. String in python are ……………..
   a) **immutable** b) changeable c) countable d) flexible
4. _____is known as slicing operator
   a) ( ) b) {} **c) [ ]** d) + =
5. Which of the following formatting character is used to print exponential notation in upper case?
   a) %e **b) %E** c) %g d) %n
6. The subscript of a string may be……………………………?
   a) Positive b) negative c) both (a) and (b) **d) either (a) or (b)**
7. Which of the following is used as place holders or replacement fields which get replaced along with format () function?
   a) { } b) <> c) ++ d) ^^
8. _________starts with a backslash and can be interpreted differently
   a) **Escape sequences** b) Capitalize c) End d) Range
9. ________returns the length of the string ?
   a) **len ( )** b) title ( ) c) upper( ) d) lower ( )
10. _______is used to add more string at the end of an existing string?
    a) Concatenation b) Repeating **c) Append** d) String slicing

PART II

Very Short Answers:

1. Define string.

String is a data type in python, which is used to handle array of characters. String is a sequence of Unicode characters that may be a combination of letters, numbers, or special symbols enclosed within single, double or even triple quotes.

Example:

‘Welcome to learning Python’
“Welcome to learning Python”
‘‘‘ “Welcome to learning Python” ’’’

2. Is it possible to modify a string in python?

It is not possible to modify a string in python since stgrings in python are immutable.

3. How will you delete a string in python?

Python will not allow deleting a particular character in a string. Whereas you can remove entire string variable using del command.

Example: Code lines to delete a string variable

```
>>> str1=”How about you”
>>> print (str1)
How about you
>>> del str1
>>> print (str1)
```

```
NameError: name ‘str1’ is not defined
```

4. What is slicing?

String slicing:

Slice is a substring of a main string. A substring can be taken from the original string by using [ ] operator and index or subscript values. Thus, [ ] is also known as slicing operator. Using slice operator, we can slice one or more substrings from a main string.

General format of slice operation:

str[start:end]

Where start is the beginning index and end is the last index value of a character in the string.

5. What will be the output of the following python code?

```
str1 =”School”
print(str1*3)
```

Output:

```
School School School
```

PART III

Short Answers:

1. Write short note on capitilize() (b) swapcase()

Syntax	Description	Example

capitalize( )	Used to capitalize the first character of the string	>>> city="chennai"
		>>> print(city.capitalize())
		Chennai

swapcase( )	It will change case of every character to its opposite case vice-versa.	>>> str1="tAmiL NaDu"
		>>> print(str1.swapcase())
		TaMIl nAdU

2. Write short notes on format ( ) function with example

The format( ) function used with strings is very versatile and powerful function used for formatting strings. The curly braces { } are used as placeholders or replacement fields which get replaced along with format( ) function.

Example:

```
num1 = int (input (“Number 1: “))
num2 = int (input (“Number 2: “))
print (“The sum of { } and { } is { }”.format (num1, num2,(num1 + num2)))
```

Output:

```
Number 1 : 34
Number 2 : 54
The sum of 34 and 54 is 88.
```

3. Write short notes on count() function in python.

Syntax : count(str, beg, end)

Description : Returns the number of substrings occurs within the given range. Remember that substring may be a single character. Range (beg and end) arguments are optional. If it is not given, python searched in whole string. Search is case sensitive.

Example:

```
>>> str1="Raja Raja Chozhan"
>>> print(str1.count('Raja'))
2
```

```
>>> print(str1.count('r'))
0
```

4. Explain any six Escape sequences supported by python.

Escape Sequence	Description
\newline	Backslash and newline ignored
\	Backslash
\’	Single quote
\f	ASCII Form feed
\n	ASCII Linefeed
\r	ASCII Carriage Return

5. What is Concatenation? Give an example

Concatenation (+)

Joining of two or more strings is called as Concatenation. The plus (+) operator is used to concatenate strings in python.

Example

```
>>> "welcome" + "Python"
'welcomePython'
```

PART IV

Long Answers:

1. Explain the string operators in python with suitable example.

String Operators:

Python provides the following operators for string operations. These operators are useful to manipulate string.

(i) Concatenation (+):

Joining of two or more strings is called as Concatenation. The plus (+) operator is used to concatenate strings in python.

Example:

```
>>> “welcome” + “Python”
‘welcomePython’
```

(ii) Append (+=):

Adding more strings at the end of an existing string is known as append. The operator += is used to append a new string with an existing string.

Example:

```
>>> str1 =”Welcome to ”
>>> str1+=”Leam Python”
>>> print (str1)
Welcome to Learn Python
```

(iii) Repeating (*):

The multiplication operator (*) is used to display a string in multiple number of times.

Example:

```
>>> str1 =”Welcome”>>> print (str1*4)
Welcome Welcome Welcome Welcome
```

(iv) String slicing:

Slice is a substring of a main string. A substring can be taken from the original string by using [ ] operator and index or subscript values. Thus, [ ] is also known as slicing operator. Using slice operator, you have to slice one or more substrings from a main string.

General format of slice operation:

str[start: end]

Where start is the beginning index and end is the last index value of a character in the string. Python takes the end value less than one from the actual index specified. For example, if you want to slice first 4 characters from a string, you have to specify it as 0 to 5. Because, python consider only the end value as n – 1.

Example:

(i) slice a single character from a string

```
>>> str1=”THIRUKKURAL ”
>>> print (str1 [0])
T .
```

(v) Stride when slicing string

When the slicing operation, you can specify a third argument as the stride, which refers to the number of characters to move forward after the first character is retrieved from the string. The default value of stride is 1.

Example:

```
>>> str1= “Welcome to learn Python”
>>> print (str1 [10:16])
learn
```

Note: Remember that, python takes the last value as n – 1

You can also use negative value as stride (third argument). If you specify a negative value, it prints in reverse order.

Example:

```
>>> str1 = “Welcome to learn Python”
>>> print(str1 [::-2])
nhy re teoIW
```

2. Write a program to check whether the give string is a palindrome or not

```
# Python Program to Check a Given String is Palindrome or Not
string = input("Please enter your own String : ")
str1 = ""
for i in string:
    str1 = i + str1
print("String in reverse Order :  ", str1)
if(string == str1):
   print("This is a Palindrome String")
else:
   print("This is Not a Palindrome String")
```

Output:

```
Please enter your own String : aabbcc
String in reverse Order :   ccbbaa
This is Not a Palindrome String
>>>
Please enter your own String : aabbaa
String in reverse Order :   aabbaa
This is a Palindrome String
```

Prepared By
S.Gowri Raamalakshmi M.C.A., B.Ed
grlakshmi30@gmail.com