EXPERIMENT - 1

OBJECTIVE:

Write a program to demonstrate different number data types in Python.

SOURCECODE:

```
a = 5
print(a, "is of type", type(a))

a = 2.0
print(a, "is of type", type(a))

a = 1+2j
print(a, "is complex number?", isinstance(1+2j,complex))
```

INPUT ANDOUTPUT:

```
5 is of type <class 'int'>
2.0 is of type <class 'float'>
(1+2j) is complex number? True
```

5 is of type <class 'int'>

2.0 is of type <class 'float'>

(1+2j) is complex number? True

VIVA Questions

1) What is Python? What are the benefits of using Python?
Python is a programming language with objects, modules, threads, exceptions and automatic memory management. The benefits of pythons are that it is simple and easy, portable, extensible, build-in data structure and it is an open source.

2) What is PEP 8?
PEP 8 is a coding convention, a set of recommendation, about how to write your Python code more readable.

3) What is pickling and unpickling?
Pickle module accepts any Python object and converts it into a string representation and dumps it into a file by using dump function, this process is called pickling. While the process of retrieving original Python objects from the stored string representation is called unpickling.

4) How Python is interpreted?
Python language is an interpreted language. Python program runs directly from the source code. It converts the source code that is written by the programmer into an intermediate language, which is again translated into machine language that has to be executed.

5) How memory is managed in Python?

- Python memory is managed by Python private heap space. All Python objects and data structures are located in a private heap. The programmer does not have an access to this private heap and interpreter takes care of this Python private heap.
- The allocation of Python heap space for Python objects is done by Python memory manager. The core API gives access to some tools for the programmer to code.
- Python also have an inbuilt garbage collector, which recycle all the unused memory and frees the memory and makes it available to the heap space.

6) What are the tools that help to find bugs or perform static analysis?
PyChecker is a static analysis tool that detects the bugs in Python source code and warns about the style and complexity of the bug. Pylint is another tool that verifies whether the module meets the coding standard.

7) What are Python decorators?
A Python decorator is a specific change that we make in Python syntax to alter functions easily.

8) What is the difference between list and tuple?
The difference between list and tuple is that list is mutable while tuple is not. Tuple can be hashed for e.g as a key for dictionaries.

9) How are arguments passed by value or by reference?
Everything in Python is an object and all variables hold references to the objects. The references values are according to the functions; as a result you cannot change the value of the references. However, you can change the objects if it is mutable.

10) What is Dict and List comprehensions are?
They are syntax constructions to ease the creation of a Dictionary or List based on existing iterable.

11) What are the built-in type does python provides?
There are mutable and Immutable types of Pythons built in types Mutable built-in types

- List
- Sets
- Dictionaries

Immutable built-in types

- Strings
- Tuples

- Numbers

12) What is namespace in Python?

In Python, every name introduced has a place where it lives and can be hooked for. This is known as namespace. It is like a box where a variable name is mapped to the object placed. Whenever the variable is searched out, this box will be searched, to get corresponding object.

13) What is lambda in Python?

It is a single expression anonymous function often used as inline function.

14) Why lambda forms in python does not have statements?

A lambda form in python does not have statements as it is used to make new function object and then return them at runtime.

15) What is pass in Python?

Pass means, no-operation Python statement, or in other words it is a place holder in compound statement, where there should be a blank left and nothing has to be written there.

16) In Python what are iterators?

In Python, iterators are used to iterate a group of elements, containers like list.

17) What is unittest in Python?

A unit testing framework in Python is known as unittest. It supports sharing of setups, automation testing, shutdown code for tests, aggregation of tests into collections etc.

18) In Python what is slicing?

A mechanism to select a range of items from sequence types like list, tuple, strings etc. is known as slicing.

19) What are generators in Python?

The way of implementing iterators are known as generators. It is a normal function except that it yields expression in the function.

20) What is docstring in Python?

A Python documentation string is known as docstring, it is a way of documenting Python functions, modules and classes.

21) How can you copy an object in Python?

To copy an object in Python, you can try copy.copy () or copy.deepcopy() for the general case. You cannot copy all objects but most of them.

22) What is negative index in Python?

Python sequences can be index in positive and negative numbers. For positive index, 0 is the first index, 1 is the second index and so forth. For negative index, (-1) is the last index and (-2) is the second last index and so forth.

23) How you can convert a number to a string?

In order to convert a number into a string, use the inbuilt function str(). If you want a octal or hexadecimal representation, use the inbuilt function oct() or hex().

24) What is the difference between Xrange and range?

Xrange returns the xrange object while range returns the list, and uses the same memory and no matter what the range size is.

25) What is module and package in Python?

In Python, module is the way to structure program. Each Python program file is a module, which imports other modules like objects and attributes.

The folder of Python program is a package of modules. A package can have modules or subfolders.

26) Mention what are the rules for local and global variables in Python?

Local variables: If a variable is assigned a new value anywhere within the function's body, it's assumed to be local.

Global variables: Those variables that are only referenced inside a function are implicitly global.

27) How can you share global variables across modules?

To share global variables across modules within a single program, create a special module. Import the config module in all modules of your application. The module will be available as a global variable across modules.

28) Explain how can you make a Python Script executable on Unix?

To make a Python Script executable on Unix, you need to do two things,

- Script file's mode must be executable and
- the first line must begin with # (#!/usr/local/bin/python)

29) Explain how to delete a file in Python?

By using a command os.remove (filename) or os.unlink(filename)

30) Explain how can you generate random numbers in Python?

To generate random numbers in Python, you need to import command as

import random

random.random()

This returns a random floating point number in the range [0,1)

EXPERIMENT - 2

OBJECTIVE:

Write a program to perform different Arithmetic Operations on numbers in Python.

SOURCECODE:

```
x = 15
y = 4

# Output: x + y = 19
print('x + y =',x+y)

# Output: x - y = 11
print('x - y =',x-y)

# Output: x * y = 60
print('x * y =',x*y)

# Output: x / y = 3.75
print('x / y =',x/y)

# Output: x // y = 3
print('x // y =',x//y)

# Output: x ** y = 50625
print('x ** y =',x**y)
```

INPUT ANDOUTPUT:

```
x + y = 19
x - y = 11
x * y = 60
```

```
x / y = 3.75
x // y = 3
x ** y = 50625
```

VIVA Questions

1. Which of these in not a core data type?
a) Lists
b) Dictionary
c) Tuples
d) Class
Answer: d
Explanation: Class is a user defined data type.
2. Given a function that does not return any value, What value is thrown by default when executed in shell.
a) int
b) bool
c) void
d) None
Answer: d
Explanation: Python shell throws a NoneType object back.
3. Following set of commands are executed in shell, what will be the output?

```
>>>str="hello"
>>>str[:2]
>>>
```

a) he
b) lo
c) olleh
d) hello
Answer: a
Explanation: We are printing only the 1st two bytes of string and hence the answer is “he”.
4. Which of the following will run without errors ?
a) round(45.8)
b) round(6352.898,2,5)
c) round()
d) round(7463.123,2,1)
Answer: a
Explanation: Execute help(round) in the shell to get details of the parameters that are passed into the round function.
5. What is the return type of function id?
a) int
b) float
c) bool
d) dict
Answer: a
Explanation: Execute help(id) to find out details in python shell.id returns a integer value that is unique.
6. In python we do not specify types,it is directly interpreted by the compiler, so consider the following operation to be performed.

```
>>>x =13?2
```

objective is to make sure x has a integer value, select all that apply (python 3.xx)
a) x = 13 // 2
b) x = int(13 / 2)
c) x = 13 % 2
d) All of the mentioned
Answer: d
Explanation: // is integer operation in python 3.0 and int(..) is a type cast operator.
7. What error occurs when you execute?
apple = mango
a) SyntaxError
b) NameError
c) ValueError
d) TypeError
Answer: b
Explanation: Mango is not defined hence name error.
8. Carefully observe the code and give the answer.

```
1. def example(a):
2.     a = a +'2'
3.      a = a*2
4. return a
5. >>>example("hello")
```

a) indentation Error
b) cannot perform mathematical operation on strings
c) hello2
d) hello2hello2
Answer: a
Explanation: Python codes have to be indented properly.
9. What data type is the object below ?
L = [1, 23, 'hello', 1].
a) list
b) dictionary
c) array
d) tuple
Answer: a
Explanation: List data type can store any values within it.
10. In order to store values in terms of key and value we use what core data type.
a) list
b) tuple
c) class
d) dictionary
Answer: d
Explanation: Dictionary stores values in terms of keys and values.
11. Which of the following results in a SyntaxError ?
a) "'Once upon a time…", she said.'
b) "He said, 'Yes!'"
c) '3\'
d) "'That's okay"'
Answer: c
Explanation: Carefully look at the colons.
advertisement
12. The following is displayed by a print function call:

```
1. tom
```

2. dick
3. harry

Select all of the function calls that result in this output
a) print("'tom
\ndick
\nharry"')
b) print("'tomdickharry"')
c) print('tom\ndick\nharry')
d) print('tom
dick
harry')
Answer: c
Explanation: The \n adds a new line.
13. What is the average value of the code that is executed below ?

1. >>>grade1 =80
2. >>>grade2 =90
3. >>>average =(grade1 + grade2)/2

a) 85.0
b) 85.1
c) 95.0
d) 95.1
Answer: a
Explanation: Cause a decimal value of 0 to appear as output.
14. Select all options that print
hello-how-are-you
a) print('hello', 'how', 'are', 'you')
b) print('hello', 'how', 'are', 'you' + '-' * 4)
c) print('hello-' + 'how-are-you')
d) print('hello' + '-' + 'how' + '-' + 'are' + 'you')
Answer: c
Explanation: Execute in the shell.
15. What is the return value of trunc() ?
a) int
b) bool
c) float
d) None
Answer: a
Explanation: Execute help(math.trunc) to get details.

16. What is the output of print 0.1 + 0.2 == 0.3?
a) True
b) False
c) Machine dependent
d) Error
Answer: b
Explanation: Neither of 0.1, 0.2 and 0.3 can be represented accurately in binary. The round off errors from 0.1 and 0.2 accumulate and hence there is a difference of 5.5511e-17 between (0.1 + 0.2) and 0.3.
17. Which of the following is not a complex number?
a) k = 2 + 3j
b) k = complex(2, 3)
c) k = 2 + 3l
d) k = 2 + 3J

Answer: c
18. What is the type of inf?
a) Boolean
b) Integer
c) Float
d) Complex
Answer: c
Explanation: Infinity is a special case of floating point numbers. It can be obtained by float('inf').
19. What does ~4 evaluate to?
a) -5
b) -4
c) -3
d) +3
Answer: a
Explanation: ~x is equivalent to -(x+1).
20. What does ~~~~~~5 evaluate to?
a) +5
b) -11
c) +11
d) -5
Answer: a
Explanation: ~x is equivalent to -(x+1).
21. Which of the following is incorrect?
a) x = 0b101
b) x = 0x4f5
c) x = 19023
d) x = 03964
Answer: d
Explanation: Numbers starting with a 0 are octal numbers but 9 isn't allowed in octal numbers.
22. What is the result of cmp(3, 1)?
a) 1
b) 0
c) True
d) False
Answer: a
Explanation: cmp(x, y) returns 1 if x > y, 0 if x == y and -1 if x < y.
23. Which of the following is incorrect?
a) float('inf')
b) float('nan')
c) float('56'+'78')
d) float('12+34′)
Answer: d
Explanation: '+' cannot be converted to a float.
24. What is the result of round(0.5) – round(-0.5)?
a) 1.0
b) 2.0
c) 0.0
d) None of the mentioned
Answer: b
Explanation: Python rounds off numbers away from 0 when the number to be rounded off is exactly halfway through. round(0.5) is 1 and round(-0.5) is -1.

25. What does 3 ^ 4 evaluate to?
a) 81
b) 12
c) 0.75
d) 7
Answer: d
Explanation: ^ is the Binary XOR operator.
26) Explain how can you access a module written in Python from C?
You can access a module written in Python from C by following method,
Module = =PyImport_ImportModule("<modulename>");
27) Mention the use of // operator in Python?
It is a Floor Divisionoperator , which is used for dividing two operands with the result as quotient showing only digits before the decimal point. For instance, 10//5 = 2 and 10.0//5.0 = 2.0.
28) Mention five benefits of using Python?

- Python comprises of a huge standard library for most Internet platforms like Email, HTML, etc.
- Python does not require explicit memory management as the interpreter itself allocates the memory to new variables and free them automatically
- Provide easy readability due to use of square brackets
- Easy-to-learn for beginners
- Having the built-in data types saves programming time and effort from declaring variables

29) Mention the use of the split functionin Python?
The use of the split function in Python is that it breaks a string into shorter strings using the defined separator. It gives a list of all words present in the string.
30) Explain what is Flask & its benefits?
Flask is a web micro framework for Python based on "Werkzeug, Jinja 2 and good intentions" BSD licensed. Werkzeug and jingja are two of its dependencies.
Flask is part of the micro-framework. Which means it will have little to no dependencies on external libraries. It makes the framework light while there is little dependency to update and less security bugs.

EXPERIMENT -3

OBJECTIVE:

Write a program to create, concatenate and print a string and accessing sub-string from a given string.

SOURCECODE:

```
# all of the following are equivalent
my_string = 'Hello'
print(my_string)
my_string = "Hello"
print(my_string)
my_string = '''Hello'''
print(my_string)
# triple quotes string can extend multiple lines
my_string = """Hello, welcome to
           the world of Python"""
print(my_string)
c=" mlritm"
print(my_string+c)
# substring function

print(my_string[5:11])
```

INPUT ANDOUTPUT:

```
Hello
Hello
Hello
Hello, welcome to
           the world of Python
Hello, welcome to
           the world of Python mlritm
, welc
```

VIVA Questions

1. What is the maximum possible length of an identifier?
   a) 31 characters
   b) 63 characters
   c) 79 characters
   d) none of the mentioned

Answer: d
Explanation: Identifiers can be of any length.

2. Which of the following is invalid?
   a) _a = 1
   b) __a = 1
   c) __str__ = 1
   d) none of the mentioned

   Answer: d
   Explanation: All the statements will execute successfully but at the cost of reduced readability.

3. Which of the following is an invalid variable?

   a) my_string_1

   b) 1st_string

   c) foo

   d) _

   Answer: b
   Explanation: Variable names should not start with a number.

4. Why are local variable names beginning with an underscore discouraged?
   a) they are used to indicate a private variables of a class
   b) they confuse the interpreter
   c) they are used to indicate global variables
   d) they slow down execution

   Answer: a
   Explanation: As Python has no concept of private variables, leading underscores are used to indicate variables that must not be accessed from outside the class.

5. Which of the following is not a keyword?

   a) eval

   b) assert

   c) nonlocal

   d) pass

   Answer: a
   Explanation: eval can be used as a variable.

6. All keywords in Python are in
   a) lower case
   b) UPPER CASE
   c) Capitalized
   d) None of the mentioned
   Answer: d
   Explanation: True, False and None are capitalized while the others are in lower case.

7. Which of the following is true for variable names in Python?
   a) unlimited length
   b) all private members must have leading and trailing underscores
   c) underscore and ampersand are the only two special characters allowed
   d) none of the mentioned
   Answer: a
   Explanation: Variable names can be of any length.

8. Which of the following is an invalid statement?

   a) abc = 1,000,000

   b) a b c = 1000 2000 3000

c) a,b,c = 1000, 2000, 3000

d) a_b_c = 1,000,000

Answer: b
Explanation: Spaces are not allowed in variable names.

9. Which of the following cannot be a variable?
   a) __init__
   b) in
   c) it
   d) on

   Answer: b
   Explanation: in is a keyword.

10. Is Python case sensitive when dealing with identifiers?
    a) yes
    b) no
    c) machine dependent
    d) none of the mentioned
    Answer: a
    Explanation: Case is always significant.

11. The value of the expressions 4/(3*(2-1)) and 4/3*(2-1) is the same. State whether true or false.
    a) True
    b) False

    Answer: a
    Explanation: Although the presence of parenthesis does affect the order of precedence, in the case shown above, it is not making a difference. The result of both of these expressions is 1.333333333. Hence the statement is true.

12. The value of the expression:
    4 + 3 % 5
    Explanation: The order of precedence is: %, +. Hence the expression above, on simplification results in 4 + 3 = 7. Hence the result is 7.

13. Evaluate the expression given below if A= 16 and B = 15.
    A % B // A
    Answer: b
    Explanation: The above expression is evaluated as: 16%15//16, which is equal to 1//16, which results in 0.

14. Which of the following operators has its associativity from right to left?
    a) +
    b) //
    c) %
    d) **

    Answer: d
    Explanation: All of the operators shown above have associativity from left to right, except exponentiation operator (**) which has its associativity from right to left.

15. What is the value of x if:
    x = int(43.55+2/2)
    Explanation: The expression shown above is an example of explicit conversion. It is evaluated as int(43.55+1) = int(44.55) = 44. Hence the result of this expression is 44.

16. What is the value of the following expression?
    2+4.00, 2**4.0
    Explanation: The result of the expression shown above is (6.0, 16.0). This is because the result is automatically rounded off to one decimal place.

17. Which of the following is the truncation division operator?
    a) /
    b) %

c) //
d) |
Answer: c
Explanation: // is the operator for truncation division. It it called so because it returns only the integer part of the quotient, truncating the decimal part. For example: 20//3 = 6.

18. What are the values of the following expressions:
2**(3**2)
(2**3)**2
2**3**2
Answer: d
Explanation: Expression 1 is evaluated as: 2**9, which is equal to 512.Expression 2 is evaluated as 8**2, which is equal to 64. The last expression is evaluated as 2**(3**2). This is because the associativity of ** operator is from right to left. Hence the result of the third expression is 512.

19. What is the value of the following expression:
8/4/2, 8/(4/2)
Answer: a
Explanation: The above expressions are evaluated as: 2/2, 8/2, which is equal to (1.0, 4.0).

20. What is the value of the following expression:
float(22//3+3/3)
Answer: b
Explanation: The expression shown above is evaluated as: float(7+1) = float(8) = 8.0. Hence the result of this expression is 8.0.

21. What is the result of the snippet of code shown below if x=1?
x<<2
Explanation: The binary form of 1 is 0001. The expression x<<2 implies we are performing bitwise left shift on x. This shift yields the value: 0100, which is the binary form of the number 4.

22. The output of the expression is:
bin(29)
'0b10111'
b) '0b11101'
c) '0b11111'
d) '0b11011'

Answer: b
Explanation: The binary form of the number 29 is 11101. Hence the output of this expression is '0b11101'.

23. What is the value of x if:
x>>2=2
Explanation: When the value of x is equal to 8 (1000), then x>>2 (bitwise right shift) yields the value 0010, which is equal to 2. Hence the value of x is 8.

24. What is the result of the expression:
int(1011)?
Explanation: The result of the expression shown will be 1011. This is because we have not specified the base in this expression. Hence it automatical

25. To find the decimal value of 1111, that is 15, we can use the function:

Explanation: The expression int('1111',2) gives the result 15. The expression int('1111', 10) will give the result 1111.

26. What is the result of the expression if x=15 and y=12:
x & y
Explanation: The symbol '&' represents bitwise AND. This gives 1 if both the bits are equal to 1, else it gives 0. The binary form of 15 is 1111 and that of 12 is 1100. Hence on performing the bitwise AND operation, we get 1100, which is equal to 12.
27. Which of the following expressions results in an error?
a) int(1011)

b) int('1011',23)
c) int(1011,2)
d) int('1011')

Answer: c
Explanation: The expression int(1011,2) results in an error. Had we written this expression as int('1011',2), then there would not be an error.
advertisement

28. Which of the following represents the bitwise XOR operator?

a) &

b) ^

c) |

d) !

Answer: b
Explanation: The ^ operator represent bitwise XOR operation. &: bitwise AND, | : bitwise OR and ! represents bitwise NOT.

29. What is the value of this expression?
bin(0x8)
Explanation: The prefix 0x specifies that the value is hexadecimal in nature. When we convert this hexadecimal value to binary form, we get the result as: '0b1000'.

30. What is the result of the expression:
0x35 | 0x75
Explanation: The binary value of 0x35 is 110101 and that of 0x75 is 1110101. On OR-ing these two values we get the output as: 1110101, which is equal to 117. Hence the result of the above expression is 117.

EXPERIMENT -4

OBJECTIVE:

Write a python script to print the current date in the following format "Sun May 29

02:26:23 IST 2017”

SOURCECODE:

```
from datetime import date

today =date.today()

# dd/mm/YY
d1 =today.strftime("%d/%m/%Y")
print("d1 =", d1)

# Textual month, day and year
d2 =today.strftime("%B %d, %Y")
print("d2 =", d2)

# mm/dd/y
d3 =today.strftime("%m/%d/%y")
print("d3 =", d3)

# Month abbreviation, day and year
d4 =today.strftime("%b-%d-%Y")
print("d4 =", d3)
```

INPUT ANDOUTPUT:

```
d1 = 25/12/2018
d2 = December 25, 2018
d3 = 12/25/18
d4 = 12/25/18
```

VIVA Questions

1. What is Python?
   Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.
2. What is the purpose of PYTHONPATH environment variable?

PYTHONPATH - It has a role similar to PATH. This variable tells the Python interpreter where to locate the module files imported into a program. It should include the Python source library directory and the directories containing Python source code. PYTHONPATH is sometimes preset by the Python installer.

3. What is the purpose of PYTHONSTARTUP environment variable?
   PYTHONSTARTUP - It contains the path of an initialization file containing Python source code. It is executed every time you start the interpreter. It is named as .pythonrc.py in Unix and it contains commands that load utilities or modify PYTHONPATH.
4. What is the purpose of PYTHONCASEOK environment variable?
   PYTHONCASEOK − It is used in Windows to instruct Python to find the first case-insensitive match in an import statement. Set this variable to any value to activate it.
5. What is the purpose of PYTHONHOME environment variable?
   PYTHONHOME − It is an alternative module search path. It is usually embedded in the PYTHONSTARTUP or PYTHONPATH directories to make switching module libraries easy.
6. Is python a case sensitive language?
   Yes! Python is a case sensitive programming language.
7. What are the supported data types in Python?
   Python has five standard data types −

   Numbers

   String

   List

   Tuple

   Dictionary

8. What is the output of print str if str = 'Hello World!'?
   It will print complete string. Output would be Hello World!.
9. What is the output of print str[0] if str = 'Hello World!'?
   It will print first character of the string. Output would be H.
10. What is the output of print str[2:5] if str = 'Hello World!'?
    It will print characters starting from 3rd to 5th. Output would be llo.
11. What is the output of print str[2:] if str = 'Hello World!'?
    It will print characters starting from 3rd character. Output would be lloWorld!.
12. What is the output of print str * 2 if str = 'Hello World!'?
    It will print string two times. Output would be Hello World!Hello World!.
13. What is the output of print str + "TEST" if str = 'Hello World!'?
    It will print concatenated string. Output would be Hello World!TEST.
14. What is the output of print list if list = ['abcd', 786 , 2.23, 'john', 70.2]?
    It will print complete list. Output would be ['abcd', 786, 2.23, 'john', 70.200000000000003].
15. What is the output of print list[0] if list = ['abcd', 786 , 2.23, 'john', 70.2]?
    It will print first element of the list. Output would be abcd.
16. What is the output of print list[1:3] if list = ['abcd', 786 , 2.23, 'john', 70.2]?
    It will print elements starting from 2nd till 3rd. Output would be [786, 2.23].
17. What is the output of print list[2:] if list = ['abcd', 786 , 2.23, 'john', 70.2]?

It will print elements starting from 3rd element. Output would be [2.23, 'john', 70.200000000000003].

18. What is the output of print tinylist * 2 if tinylist = [123, 'john']?
    It will print list two times. Output would be [123, 'john', 123, 'john'].
19. What is the output of print list1 + list2, if list1 = ['abcd', 786 , 2.23, 'john', 70.2] and ist2 = [123, 'john']?
    It will print concatenated lists. Output would be ['abcd', 786, 2.23, 'john', 70.2, 123, 'john']
20. What are tuples in Python?
    A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.
21. What is the difference between tuples and lists in Python?
    The main differences between lists and tuples are − Lists are enclosed in brackets ([ ]) and their elements and size can be changed, while tuples are enclosed in parentheses (( )) and cannot be updated. Tuples can be thought of as read-only lists.
22. What is the output of print tuple if tuple = ('abcd', 786 , 2.23, 'john', 70.2)?
    It will print complete tuple. Output would be ('abcd', 786, 2.23, 'john', 70.200000000000003).
23. What is the output of print tuple[0] if tuple = ('abcd', 786 , 2.23, 'john', 70.2)?
    It will print first element of the tuple. Output would be abcd.
24. What is the output of print tuple[1:3] if tuple = ('abcd', 786 , 2.23, 'john', 70.2)?
    It will print elements starting from 2nd till 3rd. Output would be (786, 2.23).
25. What is the output of print tuple[2:] if tuple = ('abcd', 786 , 2.23, 'john', 70.2)?
    It will print elements starting from 3rd element. Output would be (2.23, 'john', 70.200000000000003).
26. What is the output of print tinytuple * 2 if tinytuple = (123, 'john')?
    It will print tuple two times. Output would be (123, 'john', 123, 'john').
27. What is the output of print tuple + tinytuple if tuple = ('abcd', 786 , 2.23, 'john', 70.2) and tinytuple = (123, 'john')?
    It will print concatenated tuples. Output would be ('abcd', 786, 2.23, 'john', 70.200000000000003, 123, 'john').
28. What are Python's dictionaries?
    Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.
29. How will you get all the keys from the dictionary?
    Using dictionary.keys() function, we can get all the keys from the dictionary object.

```
printdict.keys()# Prints all the keys
```

30. How will you get all the values from the dictionary?
    Using dictionary.values() function, we can get all the values from the dictionary object.

```
printdict.values()# Prints all the values
```

EXPERIMENT -5

OBJECTIVE:

Write a program to create, append, and remove lists in python.

SOURCECODE:

```
my_list = ['p','r','o','b','l','e','m']
my_list.remove('p')

# Output: ['r', 'o', 'b', 'l', 'e', 'm']
print(my_list)

# Output: 'o'
print(my_list.pop(1))

# Output: ['r', 'b', 'l', 'e', 'm']
print(my_list)

# Output: 'm'
print(my_list.pop())

# Output: ['r', 'b', 'l', 'e']
print(my_list)

my_list.clear()

# Output: []
print(my_list)
```

INPUT ANDOUTPUT:

```
my_list=['p','r','o','b','l','e','m']
>>>my_list[2:3]=[]
>>>my_list
['p','r','b','l','e','m']
>>>my_list[2:5]=[]
>>>my_list
['p','r','m']
```

VIVA Questions

1. How will you convert a string to an int in python?
   int(x [,base]) - Converts x to an integer. base specifies the base if x is a string.
2. How will you convert a string to a long in python?

long(x [,base]) - Converts x to a long integer. base specifies the base if x is a string.

3. How will you convert a string to a float in python?
   float(x) − Converts x to a floating-point number.
4. How will you convert a object to a string in python?
   str(x) − Converts object x to a string representation.
5. How will you convert a object to a regular expression in python?
   repr(x) − Converts object x to an expression string.
6. How will you convert a String to an object in python?
   eval(str) − Evaluates a string and returns an object.
7. How will you convert a string to a tuple in python?
   tuple(s) − Converts s to a tuple.
8. How will you convert a string to a list in python?
   list(s) − Converts s to a list.
9. How will you convert a string to a set in python?
   set(s) − Converts s to a set.
10. How will you create a dictionary using tuples in python?
    dict(d) − Creates a dictionary. d must be a sequence of (key,value) tuples.
11. How will you convert a string to a frozen set in python?
    frozenset(s) − Converts s to a frozen set.
12. How will you convert an integer to a character in python?
    chr(x) − Converts an integer to a character.
13. How will you convert an integer to an unicode character in python?
    unichr(x) − Converts an integer to a Unicode character.
14. How will you convert a single character to its integer value in python?
    ord(x) − Converts a single character to its integer value.
15. How will you convert an integer to hexadecimal string in python?
    hex(x) − Converts an integer to a hexadecimal string.
16. How will you convert an integer to octal string in python?
    oct(x) − Converts an integer to an octal string.
17. What is the purpose of ** operator?
    ** Exponent − Performs exponential (power) calculation on operators. a**b = 10 to the power 20 if a = 10 and b = 20.
18. What is the purpose of // operator?
    // Floor Division − The division of operands where the result is the quotient in which the digits after the decimal point are removed.
19. What is the purpose of is operator?
    is − Evaluates to true if the variables on either side of the operator point to the same object and false otherwise. x is y, here is results in 1 if id(x) equals id(y).
20. What is the purpose of not in operator?
    not in − Evaluates to true if it does not finds a variable in the specified sequence and false otherwise. x not in y, here not in results in a 1 if x is not a member of sequence y.
21. What is the purpose break statement in python?
    break statement − Terminates the loop statement and transfers execution to the statement immediately following the loop.
22. What is the purpose continue statement in python?
    continue statement − Causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating.

23. What is the purpose pass statement in python?
    pass statement − The pass statement in Python is used when a statement is required syntactically but you do not want any command or code to execute.
24. How can you pick a random item from a list or tuple?
    choice(seq) − Returns a random item from a list, tuple, or string.
25. How can you pick a random item from a range?
    randrange ([start,] stop [,step]) − returns a randomly selected element from range(start, stop, step).
26. How can you get a random number in python?
    random() − returns a random float r, such that 0 is less than or equal to r and r is less than 1.
27. How will you set the starting value in generating random numbers?
    seed([x]) − Sets the integer starting value used in generating random numbers. Call this function before calling any other random module function. Returns None.
28. How will you randomizes the items of a list in place?
    shuffle(lst) − Randomizes the items of a list in place. Returns None.
29. How will you capitalizes first letter of string?
    capitalize() − Capitalizes first letter of string.
30. How will you check in a string that all characters are alphanumeric?
    isalnum() − Returns true if string has at least 1 character and all characters are alphanumeric and false otherwise.

EXPERIMENT -6

OBJECTIVE:

Write a program to demonstrate working with tuples in python.

SOURCECODE:

```
        # empty tuple
        # Output: ()
        my_tuple = ()
        print(my_tuple)

        # tuple having integers
        # Output: (1, 2, 3)
        my_tuple = (1, 2, 3)
        print(my_tuple)

        # tuple with mixed datatypes
        # Output: (1, "Hello", 3.4)
        my_tuple = (1, "Hello", 3.4)
        print(my_tuple)

        # nested tuple
        # Output: ("mouse", [8, 4, 6], (1, 2, 3))
        my_tuple = ("mouse", [8, 4, 6], (1, 2, 3))
        print(my_tuple)

        # tuple can be created without parentheses
        # also called tuple packing
        # Output: 3, 4.6, "dog"

        my_tuple = 3, 4.6, "dog"
        print(my_tuple)

        # tuple unpacking is also possible
        # Output:
        # 3
        # 4.6
        # dog
        a, b, c = my_tuple
        print(a)
        print(b)
print(c)
```

INPUT ANDOUTPUT:

()

(1, 2, 3)

(1, 'Hello', 3.4)

('mouse', [8, 4, 6], (1, 2, 3))

(3, 4.6, 'dog')

3

4.6

dog

VIVA Questions

1. How will you check in a string that all characters are digits?
   isdigit() − Returns true if string contains only digits and false otherwise.
2. How will you check in a string that all characters are in lowercase?

islower() − Returns true if string has at least 1 cased character and all cased characters are in lowercase and false otherwise.

3. How will you check in a string that all characters are numerics?
   isnumeric() − Returns true if a unicode string contains only numeric characters and false otherwise.
4. How will you check in a string that all characters are whitespaces?
   isspace() − Returns true if string contains only whitespace characters and false otherwise.
5. How will you check in a string that it is properly titlecased?
   istitle() − Returns true if string is properly "titlecased" and false otherwise.
6. How will you check in a string that all characters are in uppercase?
   isupper() − Returns true if string has at least one cased character and all cased characters are in uppercase and false otherwise.
7. How will you merge elements in a sequence?
   join(seq) − Merges (concatenates) the string representations of elements in sequence seq into a string, with separator string.
8. How will you get the length of the string?
   len(string) − Returns the length of the string.
9. How will you get a space-padded string with the original string left-justified to a total of width columns?
   ljust(width[, fillchar]) − Returns a space-padded string with the original string left-justified to a total of width columns.
10. How will you convert a string to all lowercase?
    lower() − Converts all uppercase letters in string to lowercase.
11. How will you remove all leading whitespace in string?
    lstrip() − Removes all leading whitespace in string.
12. How will you get the max alphabetical character from the string?
    max(str) − Returns the max alphabetical character from the string str.
13. How will you get the min alphabetical character from the string?
    min(str) − Returns the min alphabetical character from the string str.
14. How will you replaces all occurrences of old substring in string with new string?
    replace(old, new [, max]) − Replaces all occurrences of old in string with new or at most max occurrences if max given.
15. How will you remove all leading and trailing whitespace in string?
    strip([chars]) − Performs both lstrip() and rstrip() on string.
16. How will you change case for all letters in string?
    swapcase() − Inverts case for all letters in string.
17. How will you get titlecased version of string?
    title() − Returns "titlecased" version of string, that is, all words begin with uppercase and the rest are lowercase.
18. How will you convert a string to all uppercase?
    upper() − Converts all lowercase letters in string to uppercase.
19. How will you check in a string that all characters are decimal?
    isdecimal() − Returns true if a unicode string contains only decimal characters and false otherwise.
20. What is the difference between del() and remove() methods of list?
    To remove a list element, you can use either the del statement if you know exactly which element(s) you are deleting or the remove() method if you do not know.

21. What is the output of len([1, 2, 3])?
    3.
22. What is the output of [1, 2, 3] + [4, 5, 6]?
    [1, 2, 3, 4, 5, 6]
23. What is the output of ['Hi!'] * 4?
    ['Hi!', 'Hi!', 'Hi!', 'Hi!']
24. What is the output of 3 in [1, 2, 3]?
    True
25. What is the output of for x in [1, 2, 3]: print x?

```
1
2
3
```

26. What is the output of L[2] if L = [1,2,3]?
    3, Offsets start at zero.
27. What is the output of L[-2] if L = [1,2,3]?
    1, Negative: count from the right.
28. What is the output of L[1:] if L = [1,2,3]?
    2, 3, Slicing fetches sections.
29. How will you compare two lists?
    cmp(list1, list2) − Compares elements of both lists.
30. How will you get the length of a list?
    len(list) − Gives the total length of the list.

EXPERIMENT -7

OBJECTIVE:

Write a program to demonstrate working with dictionaries in python.

SOURCECODE:

```
my_dict = {'name':'Jack', 'age': 26}

# Output: Jack
print(my_dict['name'])

# Output: 26
print(my_dict.get('age'))

# Trying to access keys which doesn't exist throws error
# my_dict.get('address')
# my_dict['address']
```

INPUT ANDOUTPUT:

```
Jack
26
```

VIVA Questions

1. How will you get the max valued item of a list?
   max(list) − Returns item from the list with max value.
2. How will you get the min valued item of a list?

min(list) − Returns item from the list with min value.

3. How will you get the index of an object in a list?
   list.index(obj) − Returns the lowest index in list that obj appears.
4. How will you insert an object at given index in a list?
   list.insert(index, obj) − Inserts object obj into list at offset index.
5. How will you remove last object from a list?
   list.pop(obj=list[-1]) − Removes and returns last object or obj from list.
6. How will you remove an object from a list?
   list.remove(obj) − Removes object obj from list.
7. How will you reverse a list?
   list.reverse() − Reverses objects of list in place.
8. How will you sort a list?
   list.sort([func]) − Sorts objects of list, use compare func if given.
9. Is Python platform independent?
   No
   There are some modules and functions in python that can only run on certain platforms.
10. Do you think Python has a complier?
    Yes it has a complier which works automatically so we don't notice the compiler of python.
11. Which programming Language is an implementation of Python programming language designed to run on Java Platform?Is there any double data type in Python?
    No
12. Is String in Python are immutable? (Yes/No)
    Yes.
13. Can True = False be possible in Python?
    No.
14. Which module of python is used to apply the methods related to OS.?
    OS.
15. When does a new block begin in python?
    A block begins when the line is intended by 4 spaces.
16. Name the python Library used for Machine learning.
    Scikit-learn python Library used for Machine learning
17. What does pass operation do?
    Pass indicates that nothing is to be done i.e. it signifies a no operation.
18. Name the tools which python uses to find bugs (if any).
    Pylint and pychecker.

19. Does python support multiple inheritance?
    Ans: Multiple inheritance means that a class can be derived from more than one parent classes. Python does support multiple inheritance, unlike Java.
20. What is Polymorphism in Python?
    Ans: Polymorphism means the ability to take multiple forms. So, for instance, if the parent class has a method named ABC then the child class also can have a method with the same name ABC having its own parameters and variables. Python allows polymorphism.
21. Define encapsulation in Python?
    Ans: Encapsulation means binding the code and the data together. A Python class in an example of encapsulation.
22. How do you do data abstraction in Python?
    Ans: Data Abstraction is providing only the required details and hiding the implementation from the world. It can be achieved in Python by using interfaces and abstract classes.
23. .Does python make use of access specifiers?

Ans: Python does not deprive access to an instance variable or function. Python lays down the concept of prefixing the name of the variable, function or method with a single or double underscore to imitate the behavior of protected and private access specifiers.

24. What is map function in Python?

Ans: *map* function executes the function given as the first argument on all the elements of the iterable given as the second argument. If the function given takes in more than 1 arguments, then many iterables are given. #Follow the link to know more similar functions.

25. Mention what are the rules for local and global variables in Python?
    Local variables: If a variable is assigned a new value anywhere within the function's body, it's assumed to be local.
    Global variables: Those variables that are only referenced inside a function are implicitly global.
26. How can you share global variables across modules?
    To share global variables across modules within a single program, create a special module. Import the config module in all modules of your application. The module will be available as a global variable across modules.
27. Explain how can you make a Python Script executable on Unix?
    To make a Python Script executable on Unix, you need to do two things,
    Script file's mode must be executable and
    the first line must begin with # (#!/usr/local/bin/python)
28. Explain how to delete a file in Python?
    By using a command os.remove (filename) or os.unlink(filename)
29. What is unittest in Python?
    A unit testing framework in Python is known as unittest. It supports sharing of setups, automation testing, shutdown code for tests, aggregation of tests into collections etc.
30. In Python what is slicing?
    A mechanism to select a range of items from sequence types like list, tuple, strings etc. is known as slicing.

EXPERIMENT -8

OBJECTIVE:

Write a python program to find largest of three numbers.

SOURCECODE:

```
# Python program to find the largest number among the three input numbers

# change the values of num1, num2 and num3
# for a different result
num1 = 10
num2 = 14
num3 = 12

# uncomment following lines to take three numbers from user
#num1 = float(input("Enter first number: "))
#num2 = float(input("Enter second number: "))
#num3 = float(input("Enter third number: "))

if (num1 >= num2) and (num1 >= num3):
   largest = num1
elif (num2 >= num1) and (num2 >= num3):
   largest = num2
else:
   largest = num3

print("The largest number between",num1,",",num2,"and",num3,"is",largest)
```

INPUT ANDOUTPUT:

```
The largest number between 10, 14 and 12 is 14.0
```

VIVA Questions

Q-1: What Are The Different Methods To Copy An Object In Python?
There are two ways to copy objects in Python.

- **copy.copy() function**

- It makes a copy of the file from source to destination.
- It'll return a shallow copy of the parameter.

- **copy.deepcopy() function**
  - It also produces the copy of an object from the source to destination.
  - It'll return a deep copy of the parameter that you can pass to the function.

Q-2: What Is The Purpose Of Docstrings In Python?

In Python, the docstring is what we call as the docstrings. It sets a process of recording Python functions, modules, and classes.

Q-90: Which Python Function Will You Use To Convert A Number To A String?

For converting a number into a string, you can use the built-in function **str()**. If you want an octal or hexadecimal representation, use the inbuilt function **oct()** or **hex()**.

Q-3: What's The Process To Get The Home Directory Using '~' In Python?

You need to import the os module, and then just a single line would do the rest.

```
import os
print (os.path.expanduser('~'))
```

Output:

```
/home/runner
```

Q-5: What Are The Built-In Types Available In Python?

Here is the list of most commonly used built-in types that Python supports:

- **Immutable built-in datatypes of Python**
  - Numbers
  - Strings
  - Tuples
- **Mutable built-in datatypes of Python**
  - List
  - Dictionaries
  - Sets

Q-6: How To Find Bugs Or Perform Static Analysis In A Python Application?

- You can use PyChecker, which is a static analyzer. It identifies the bugs in Python project and also reveals the style and complexity related bugs.
- Another tool is Pylint, which checks whether the Python module satisfies the coding standard.

Q-7: When Is The Python Decorator Used?

Python decorator is a relative change that you do in Python syntax to adjust the functions quickly.

Q-8: What Is The Principal Difference Between A List And The Tuple?

List Vs. Tuple.

The principal difference between a list and the tuple is that the former is mutable while the tuple is not.

A tuple is allowed to be hashed, for example, using it as a key for dictionaries.

Q-9: How Does Python Handle Memory Management?

- Python uses private heaps to maintain its memory. So the heap holds all the Python objects and the data structures. This area is only accessible to the Python interpreter; programmers can't use it.
- And it's the Python memory manager that handles the Private heap. It does the required allocation of the memory for Python objects.
- Python employs a built-in garbage collector, which salvages all the unused memory and offloads it to the heap space.

Q-10: What Are The Principal Differences Between The Lambda And Def?

Lambda Vs. Def.

- Def can hold multiple expressions while lambda is a uni-expression function.
- Def generates a function and designates a name to call it later. Lambda forms a function object and returns it.
- Def can have a return statement. Lambda can't have return statements.
- Lambda supports to get used inside a list and dictionary.

Q-11: Write A Reg Expression That Confirms An Email Id Using The Python Reg Expression Module "Re"?

Python has a regular expression module "re."

Check out the **"re"** expression that can check the email id for .com and .co.in subdomain.

```
import re
print(re.search(r"[0-9a-zA-Z.]+@[a-zA-Z]+\.(com|co\.in)$","micheal.pages@mp.com"))
```

Q-12: What Do You Think Is The Output Of The Following Code Fragment? Is There Any Error In The Code?

```
list = ['a', 'b', 'c', 'd', 'e']
```

```
print (list[10:])
```

The result of the above lines of code is []. There won't be any error like an IndexError.

You should know that trying to fetch a member from the list using an index that exceeds the member count (for example, attempting to access list[10] as given in the question) would yield an IndexError. By the way, retrieving only a slice at the starting index that surpasses the no. of items in the list won't result in an IndexError. It will just return an empty list.

Q-13: Is There A Switch Or Case Statement In Python? If Not Then What Is The Reason For The Same?

No, Python does not have a Switch statement, but you can write a Switch function and then use it.

Q-14: What Is A Built-In Function That Python Uses To Iterate Over A Number Sequence?

Range() generates a list of numbers, which is used to iterate over for loops.

```
for i in range(5):
    print(i)
```

The range() function accompanies two sets of parameters.

- **range(stop)**
  - stop: It is the no. of integers to generate and starts from zero. eg.range(3) == [0, 1, 2].
- **range([start], stop[, step])**
  - Start: It is the starting no. of the sequence.
  - Stop: It specifies the upper limit of the sequence.
  - Step: It is the incrementing factor for generating the sequence.
- **Points to note:**
  - Only integer arguments are allowed.
  - Parameters can be positive or negative.
  - The **range()** function in Python starts from the zeroth index.

Q-15: What Are The Optional Statements Possible Inside A Try-Except Block In Python?

There are two optional clauses you can use in the **try-except** block.

- The "**else"** clause
  - It is useful if you want to run a piece of code when the try block doesn't create an exception.
- The **"finally"** clause
  - It is useful when you want to execute some steps which run, irrespective of whether there occurs an exception or not.

Q-16: What Is A String In Python?

A string in Python is a sequence of alpha-numeric characters. They are immutable objects. It means that they don't allow modification once they get assigned a value. Python provides several methods, such as join(), replace(), or split() to alter strings. But none of these change the original object.

Q-17: What Is Slicing In Python?

Slicing is a string operation for extracting a part of the string, or some part of a list. In Python, a string (say text) begins at index 0, and the nth character stores at position text[n-1]. Python can also perform reverse indexing, i.e., in the backward direction, with the help of negative numbers. In Python, the slice() is also a constructor function which generates a slice object. The result is a set of indices mentioned by range(start, stop, step). The slice() method allows three parameters. 1. start – starting number for the slicing to begin. 2. stop – the number which indicates the end of slicing. 3. step – the value to increment after each index (default = 1).

Q-18: What Is %S In Python?

Python has support for formatting any value into a string. It may contain quite complex expressions.

One of the common usages is to push values into a string with the %s format specifier. The formatting operation in Python has the comparable syntax as the C function printf() has.

Q-19: Is A String Immutable Or Mutable In Python?

Python strings are indeed immutable.

Let's take an example. We have an "str" variable holding a string value. We can't mutate the container, i.e., the string, but can modify what it contains that means the value of the variable.

Q-20: What Is The Index In Python?

An index is an integer data type which denotes a position within an ordered list or a string.

In Python, strings are also lists of characters. We can access them using the index which begins from zero and goes to the length minus one.

For example, in the string "Program," the indexing happens like this:

```
Program 0 1 2 3 4 5
```

Q-21: What Is Docstring In Python?

A docstring is a unique text that happens to be the first statement in the following Python constructs:

Module, Function, Class, or Method definition.

A docstring gets added to the __doc__ attribute of the string object.

Now, read some of the Python interview questions on functions.

Q-22: What Is A Function In Python Programming?
A function is an object which represents a block of code and is a reusable entity. It brings modularity to a program and a higher degree of code reusability.
Python has given us many built-in functions such as print() and provides the ability to create user-defined functions.
Q-23: How Many Basic Types Of Functions Are Available In Python?
Python gives us two basic types of functions.
1. Built-in, and
2. User-defined.
The built-in functions happen to be part of the Python language. Some of these are print(), dir(), len(), and abs() etc.
Q-24: How Do We Write A Function In Python?
We can create a Python function in the following manner.
Step-1: to begin the function, start writing with the keyword def and then mention the function name.
Step-2: We can now pass the arguments and enclose them using the parentheses. A colon, in the end, marks the end of the function header.
Step-3: After pressing an enter, we can add the desired Python statements for execution.
Q-25: What Is A Function Call Or A Callable Object In Python?
A function in Python gets treated as a callable object. It can allow some arguments and also return a value or multiple values in the form of a tuple. Apart from the function, Python has other constructs, such as classes or the class instances which fits in the same category.
Q-26: What Is The Return Keyword Used For In Python?
The purpose of a function is to receive the inputs and return some output.
The return is a Python statement which we can use in a function for sending a value back to its caller.
Q-27: What Is “Call By Value” In Python?
In call-by-value, the argument whether an expression or a value gets bound to the respective variable in the function. Python will treat that variable as local in the function-level scope. Any changes made to that variable will remain local and will not reflect outside the function.
Q-28: What Is “Call By Reference” In Python?
We use both “call-by-reference” and “pass-by-reference” interchangeably. When we pass an argument by reference, then it is available as an implicit reference to the function, rather than a simple copy. In such a case, any modification to the argument will also be visible to the caller.
This scheme also has the advantage of bringing more time and space efficiency because it leaves the need for creating local copies.
On the contrary, the disadvantage could be that a variable can get changed accidentally during a function call. Hence, the programmers need to handle in the code to avoid such uncertainty.
Q-29: What Is The Return Value Of The Trunc() Function?
The Python trunc() function performs a mathematical operation to remove the decimal values from a particular expression and provides an integer value as its output.
Q-30: Is It Mandatory For A Python Function To Return A Value?
It is not at all necessary for a function to return any value. However, if needed, we can use None as a return value.

EXPERIMENT -9

OBJECTIVE:

Write a Python program to convert temperatures to and from Celsius, Fahrenheit. [Formula : c/5 = f-32/9]

SOURCECODE:

```
# Python Program to convert temperature in celsius to fahrenheit

# change this value for a different result
celsius = 37.5

# calculate fahrenheit
fahrenheit = (celsius * 1.8) + 32
print('%0.1f degree Celsius is equal to %0.1f degree Fahrenheit' %(celsius,fahrenheit))
```

INPUT ANDOUTPUT:

```
37.5 degree Celsius is equal to 99.5 degree Fahrenheit
```

VIVA Questions

1. What Does The Continue Do In Python?

The continue is a jump statement in Python which moves the control to execute the next iteration in a loop leaving all the remaining instructions in the block unexecuted.

The continue statement is applicable for both the "while" and "for" loops.

2. What Is The Purpose Of Id() Function In Python?

   The id() is one of the built-in functions in Python.

3. What Does The *Args Do In Python?

   We use *args as a parameter in the function header. It gives us the ability to pass N (variable) number of arguments.

4. What Does The **Kwargs Do In Python?

   We can also use the **kwargs syntax in a Python function declaration. It let us

5. What Does The __ Name __ Do In Python?

   The __name__ is a unique variable. Since Python doesn't expose the main() function, so when its interpreter gets to run the script, it first executes the code which is at level 0 indentation.

6. What Is The Purpose Of "End" In Python?

   Python's print() function always prints a newline in the end. The print() function accepts an optional parameter known as the 'end.' Its value is '\n' by default. We can change the end character in a print statement with the value of our choice using this parameter.

7. When Should You Use The "Break" In Python?

   Python provides a break statement to exit from a loop. Whenever the break hits in the code, the control of the program immediately exits from the body of the loop.

   The break statement in a nested loop causes the control to exit from the inner iterative block.

8. What Is The Difference Between Pass And Continue In Python?

   The continue statement makes the loop to resume from the next iteration.

   On the contrary, the pass statement instructs to do nothing, and the remainder of the code executes as usual.

9. What Does The Len() Function Do In Python?

   In Python, the len() is a primary string function. It determines the length of an input string.

10. What Does The Chr() Function Do In Python?

    The chr() function got re-added in Python 3.2. In version 3.0, it got removed.

11. Q-42: What Does The Ord() Function Do In Python?

    The ord(char) in Python takes a string of size one and returns an integer denoting the Unicode code format of the character in case of a Unicode type object, or the value of the byte if the argument is of 8-bit string type.

12. What Is Rstrip() In Python?

    Python provides the rstrip() method which duplicates the string but leaves out the whitespace characters from the end.

13. What Is Whitespace In Python?

    Whitespace represents the characters that we use for spacing and separation.

    They possess an "empty" representation. In Python, it could be a tab or space.

14. What Is Isalpha() In Python?

    Python provides this built-in isalpha() function for the string handling purpose.

    It returns True if all characters in the string are of alphabet type, else it returns False.

15. How Do You Use The Split() Function In Python?

    Python's split() function works on strings to cut a large piece into smaller chunks, or sub-strings. We can specify a separator to start splitting, or it uses the space as one by default.

16. What Does The Join Method Do In Python?

    Python provides the join() method which works on strings, lists, and tuples. It combines them and returns a united value.

17. What Does The Title() Method Do In Python?

    Python provides the title() method to convert the first letter in each word to capital format while the rest turns to Lowercase.

18. What Makes TheCPython Different From Python?

    CPython has its core developed in C. The prefix 'C' represents this fact. It runs an interpreter loop used for translating the Python-ish code to C language.

19. Which Package Is The Fastest Form Of Python?

    PyPy provides maximum compatibility while utilizing CPython implementation for improving its performance. The tests confirmed that PyPy is nearly five times faster than the CPython. It currently supports Python 2.7.

20. What Is GIL In Python Language?

    Python supports GIL (the global interpreter lock) which is a mutex used to secure access to Python objects, synchronizing multiple threads from running the Python bytecodes at the same time.

21. How Is Python Thread Safe?

    Python ensures safe access to threads. It uses the GIL mutex to set synchronization. If a thread loses the GIL lock at any time, then you have to make the code thread-safe.

22. How Does Python Manage The Memory?

    Python implements a heap manager internally which holds all of its objects and data structures.

23. What Is A Tuple In Python?

    A tuple is a collection type data structure in Python which is immutable.

    They are similar to sequences, just like the lists. However, There are some differences between a tuple and list; the former doesn't allow modifications whereas the list does.

24. What Is A Dictionary In Python Programming?

    A dictionary is a data structure known as an associative array in Python which stores a collection of objects. The collection is a set of keys having a single associated value. We can call it a hash, a map, or a hashmap as it gets called in other programming languages.

25. What Is The Set Object In Python?

    Sets are unordered collection objects in Python. They store unique and immutable objects. Python has its implementation derived from mathematics.

26. What Is The Use Of The Dictionary In Python?

    A dictionary has a group of objects (the keys) map to another group of objects (the values). A Python dictionary represents a mapping of unique Keys to Values.

    They are mutable and hence will not change. The values associated with the keys can be of any Python types.

27. Is Python List A Linked List?

    A Python list is a variable-length array which is different from C-style linked lists.

Internally, it has a contiguous array for referencing to other objects and stores a pointer to the array variable and its length in the list head structure.

Here are some Python interview questions on classes and objects.

28. What Does The “Self” Keyword Do?

    The **self** is a Python keyword which represents a variable that holds the instance of an object.

    In almost, all the object-oriented languages, it is passed to the methods as a hidden parameter.

29. What Is The Syntax For Dictionary Comprehension In Python?

    A dictionary has the same syntax as was for the list comprehension but the difference is that it uses curly braces:

30. What Is The Syntax For List Comprehension In Python?

The signature for the list comprehension is as follows:

```
[ expression(var)forvariniterable]
```

EXPERIMENT -10

OBJECTIVE:

Write a Python program to construct the following pattern, using a nested for loop

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

SOURCECODE:

```
n=5;
for i in range(n):
    for j in range(i):
        print ('* ', end="")
print('')

for i in range(n,0,-1):
    for j in range(i):
print('* ', end="")
print('')
```

INPUT ANDOUTPUT:

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

VIVA Questions

1. What Is The Syntax For Dictionary Comprehension In Python?

A dictionary has the same syntax as was for the list comprehension but the difference is that it uses curly braces:

```
{aKey,itsValueforaKeyiniterable}
```

2. How Do You Check The Presence Of A Key In A Dictionary?

We can use Python's "in" operator to test the presence of a key inside a dict object.

3. How Do You Delete Elements Of A Dictionary In Python?

We can delete a key in a dictionary by using the del() method.

4. How Do You Add Elements To A Dictionary In Python?

We can add elements by modifying the dictionary with a fresh key and then set the value to it.

5. How Do You Traverse Through A Dictionary Object In Python?

We can use the "for" and "in" loop for traversing the dictionary object.

6. How Do You Read From A Dictionary In Python?

To fetch data from a dictionary, we can directly access using the keys. We can enclose a "key" using brackets […] after mentioning the variable name corresponding to the dictionary.

7. How Do You Create A Dictionary In Python?

Let's take the example of building site statistics. For this, we first need to break up the key-value pairs using a colon(":"). The keys should be of an immutable type, i.e., so we'll use the data-types which don't allow changes at runtime. We'll choose from an int, string, or tuple.

8. What Are Decorators In Python?

Python decorator gives us the ability to add new behavior to the given objects dynamically.

9. What Is Class In Python?
   Python supports object-oriented programming and provides almost all OOP features to use in programs.
   A Python class is a blueprint for creating the objects. It defines member variables and gets their behavior associated with them.
   We can make it by using the keyword "class." An object gets created from the constructor. This object represents the instance of the class.
10. What Are Attributes And Methods In A Python Class?
    A class is useless if it has not defined any functionality. We can do so by adding attributes. They work as containers for data and functions. We can add an attribute directly specifying inside the class body.
11. How To Assign Values For The Class Attributes At Runtime?
    We can specify the values for the attributes at runtime. We need to add an init method and pass input to object constructor. See the following example demonstrating this.
12. What Is Inheritance In Python Programming?
    Inheritance is an OOP mechanism which allows an object to access its parent class features. It carries forward the base class functionality to the child.
13. What Is Composition In Python?
    The composition is also a type of inheritance in Python. It intends to inherit from the base class but a little differently, i.e., by using an instance variable of the base class acting as a member of the derived class.
14. What Are Errors And Exceptions In Python Programs?
    Errors are coding issues in a program which may cause it to exit abnormally.

On the contrary, exceptions happen due to the occurrence of an external event which interrupts the normal flow of the program.

15. How Do You Handle Exceptions With Try/Except/Finally In Python?
    Python lay down Try, Except, Finally constructs to handle errors as well as Exceptions. We enclose the unsafe code indented under the try block. And we can keep our fall-back code inside the except block. Any instructions intended for execution last should come under the finally block.
16. How Do You Raise Exceptions For A Predefined Condition In Python?
    We can raise an exception based on some condition.
17. What Are Python Iterators?
    Iterators in Python are array-like objects which allow moving on the next element. We use them in traversing a loop, for example, in a "for" loop.
18. What Is The Difference Between An Iterator And Iterable?
    The collection type like a list, tuple, dictionary, and set are all iterable objects whereas they are also iterable containers which return an iterator while traversing.
19. What Are Python Generators?
    A Generator is a kind of function which lets us specify a function that acts like an iterator and hence can get used in a "for" loop.
20. What Are Closures In Python?
    Python closures are function objects returned by another function. We use them to eliminate code redundancy.
21. What Are Decorators In Python?
    Python decorator gives us the ability to add new behavior to the given objects dynamically. In the example below, we've written a simple example to display a message pre and post the execution of a function.
22. How Do You Create A Dictionary In Python?
    Let's take the example of building site statistics. For this, we first need to break up the key-value pairs using a colon(":"). The keys should be of an immutable type, i.e., so we'll use the data-types which don't allow changes at runtime. We'll
23. How Do You Read From A Dictionary In Python?
    To fetch data from a dictionary, we can directly access using the keys. We can enclose a "key" using brackets […] after mentioning the variable name corresponding to the dictionary.
24. How Do You Traverse Through A Dictionary Object In Python?
    We can use the "for" and "in" loop for traversing the dictionary object.
25. How Do You Add Elements To A Dictionary In Python?
    We can add elements by modifying the dictionary with a fresh key and then set the value to it.
26. How Do You Delete Elements Of A Dictionary In Python?
    We can delete a key in a dictionary by using the del() method.
27. How Do You Check The Presence Of A Key In A Dictionary?
    We can use Python's "in" operator to test the presence of a key inside a dict object.
28. What Is The Syntax For List Comprehension In Python?
    The signature for the list comprehension is as follows:
29. What Is The Syntax For Dictionary Comprehension In Python?
    A dictionary has the same syntax as was for the list comprehension but the difference is that it uses curly braces:
30. Q-80: What Is The Syntax For Generator Expression In Python?
    The syntax for generator expression matches with the list comprehension, but the difference is that it uses parenthesis:

```
( expression(var) for var in iterable )
```

EXPERIMENT -11

OBJECTIVE:

Write a Python script that prints prime numbers less than 20.

SOURCECODE:

```
r=int(input("Enter upper limit: "))
for a inrange(2,r+1):
    k=0
foriinrange(2,a//2+1):
if(a%i==0):
            k=k+1
if(k<=0):
print(a)
```

INPUT ANDOUTPUT:

```
Enter upper limit: 15
2
3
5
7
11
13
```

VIVA Questions

1. How Do You Write A Conditional Expression In Python?

   We can utilize the following single statement as a conditional expression. default_statment if Condition else another_statement

2. What Do You Know About The Python Enumerate?

   While using the iterators, sometimes we might have a use case to store the count of iterations. Python gets this task quite easy for us by giving a built-in method known as the enumerate().

   The enumerate() function attaches a counter variable to an iterable and returns it as the "enumerated" object.

3. What Is The Use Of Globals() Function In Python?

   The globals() function in Python returns the current global symbol table as a dictionary object.

   Python maintains a symbol table to keep all necessary information about a program. This info includes the names of variables, methods, and classes used by the program.

   All the information in this table remains in the global scope of the program and Python allows us to retrieve it using the globals() method.

4. Why Do You Use The Zip() Method In Python?

   The zip method lets us map the corresponding index of multiple containers so that we can use them using as a single unit.

5. What Are Class Or Static Variables In Python Programming?

   In Python, all the objects share common class or static variables.

   But the instance or non-static variables are altogether different for different objects.

6. How Does The Ternary Operator Work In Python?

   The ternary operator is an alternative for the conditional statements. It combines true or false values with a statement that you need to test.

7. What Does The "Self" Keyword Do?

   The **self** is a Python keyword which represents a variable that holds the instance of an object.

   In almost, all the object-oriented languages, it is passed to the methods as a hidden parameter.

8. Q-88: What Are The Different Methods To Copy An Object In Python?

   There are two ways to copy objects in Python.

   copy.copy() function

   It makes a copy of the file from source to destination.

   It'll return a shallow copy of the parameter.

   copy.deepcopy() function

   It also produces the copy of an object from the source to destination.

   It'll return a deep copy of the parameter that you can pass to the function.

9. What Is The Purpose Of Docstrings In Python?

   In Python, the docstring is what we call as the docstrings. It sets a process of recording Python functions, modules, and classes.

10. Which Python Function Will You Use To Convert A Number To A String?

    For converting a number into a string, you can use the built-in function **str()**. If you want an octal or hexadecimal representation, use the inbuilt function **oct()** or **hex()**.

11. How Do You Debug A Program In Python? Is It Possible To Step Through The Python Code?

    Yes, we can use the Python debugger (**pdb**) to debug any Python program. And if we start a program using **pdb**, then it let us even step through the code.

12. List Down Some Of The PDB Commands For Debugging Python Programs?

    Here are a few PDB commands to start debugging Python code.

    Add breakpoint **(b)**

    Resume execution **(c)**

    Step by step debugging **(s)**

    Move to the next line **(n)**

    List source code **(l)**

    Print an expression **(p)**

13. What Is The Command To Debug A Python Program?

    The following command helps run a Python program in debug mode.

```
$ python -m pdb python-script.py
```

15. How Do You Monitor The Code Flow Of A Program In Python?

    In Python, we can use **the sys** module's **settrace()** method to setup trace hooks and monitor the functions inside a program.

    You need to define a trace callback method and pass it to the **settrace()** function. The callback should specify three arguments as shown below.

16. Why And When Do You Use Generators In Python?

    A generator in Python is a function which returns an iterable object. We can iterate on the generator object using the **yield** keyword. But we can only do that once because their values don't persist in memory, they get the values on the fly.

    Generators give us the ability to hold the execution of a function or a step as long as we want to keep it. However, here are a few examples where it is beneficial to use generators.

17. What Does The Yield Keyword Do In Python?

    The **yield** keyword can turn any function into a generator. It works like a standard return keyword. But it'll always return a generator object. Also, a method can have multiple calls to the **yield** keyword.

18. How To Convert A List Into Other Data Types?

    Sometimes, we don't use lists as is. Instead, we have to convert them to other types.

19. How Do You Count The Occurrences Of Each Item Present In The List Without Explicitly Mentioning Them?

    Unlike sets, lists can have items with the same values.

    In Python, the list has a **count()** function which returns the occurrences of a particular item.

20. What Are Different Ways To Create An Empty NumPy Array In Python?

    There are two methods which we can apply to create empty NumPy arrays.

    The First Method To Create An Empty Array.

```
import numpy
numpy.array([])
```

    The Second Method To Create An Empty Array.

```
# Make an empty NumPy array
```

```
numpy.empty(shape=(0,0))
```

21. What is the output of print str if str = 'Hello World!'?

    It will print complete string. Output would be Hello World!.

22. What is the output of print str[0] if str = 'Hello World!'?

    It will print first character of the string. Output would be H.

23. What is the output of print str[2:5] if str = 'Hello World!'?

    It will print characters starting from 3rd to 5th. Output would be llo.

24. What is the output of print str[2:] if str = 'Hello World!'?

    It will print characters starting from 3rd character. Output would be lloWorld!.

25. What is the output of print str * 2 if str = 'Hello World!'?

    It will print string two times. Output would be Hello World!Hello World!.

26. What is the output of print str + "TEST" if str = 'Hello World!'?

    It will print concatenated string. Output would be Hello World!TEST.

27. What is the output of print list if list = ['abcd', 786 , 2.23, 'john', 70.2]?

    It will print complete list. Output would be ['abcd', 786, 2.23, 'john', 70.200000000000003].

28. What is the output of print list[0] if list = ['abcd', 786 , 2.23, 'john', 70.2]?

    It will print first element of the list. Output would be abcd.

29. What is the output of print list[1:3] if list = ['abcd', 786 , 2.23, 'john', 70.2]?

    It will print elements starting from 2nd till 3rd. Output would be [786, 2.23].

30. What is the output of print list[2:] if list = ['abcd', 786 , 2.23, 'john', 70.2]?

    It will print elements starting from 3rd element. Output would be [2.23, 'john', 70.200000000000003].

EXPERIMENT -12

OBJECTIVE:

Write a python program to find factorial of a number using Recursion.

SOURCECODE:

```
# Python program to find the factorial of a number provided by the user.

# change the value for a different result
num = 7

# uncomment to take input from the user
#num = int(input("Enter a number: "))

factorial = 1

# check if the number is negative, positive or zero
if num < 0:
print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
print("The factorial of 0 is 1")
else:
  for i in range(1,num + 1):
      factorial = factorial*i
print("The factorial of",num,"is",factorial)
```

INPUT ANDOUTPUT:

```
The factorial of 7 is 5040
```

VIVA Questions

1. **How do you debug a Python program?**
   By using this command we can debug a python program
   $ python -m pdb python-script.py

2. **What is <Yield> Keyword in Python?**

The <yield> keyword in Python can turn any function into a generator. Yields work like a standard return keyword. But it'll always return a generator object. Also, a function can have multiple calls to the <yield> keyword.

3. **How to convert a list into a string?**

When we want to convert a list into a string, we can use the <".join()> method which joins all the elements into one and returns as a string.

4. **How to convert a list into a tuple?**

By using Python <tuple()> function we can convert a list into a tuple. But we can't change the list after turning it into tuple, because it becomers immutable.

5. **How to convert a list into a set?**

User can convert list into set by using <set()> function.

6. **How to count the occurrences of a perticular element in the list?**

In Python list, we can count the occurences of a individual element by using a <count()> function.

7. **What is NumPy array?**

NumPy arrays are more flexible then lists in Python. By using NumPy arrays reading and writing items is faster and more efficient.

8. **How can you create Empty NumPy Array In Python?**

We can create Empty NumPy Array in two ways in Python,

1) import numpy
   numpy.array([])

2) numpy.empty(shape=(0,0))

9. **What is a negative index in Python?**

Python has a special feature like a negative index in Arrays and Lists. Positive index reads the elements from the starting of an array or list but in the negative index, Python reads elements from the end of an array or list.

10. **What is the output of the below code?**

```
>> import array
>>> a = [1, 2, 3]
>>> print a[-3]
>>> print a[-2]
>>> print a[-1]
```

A) The output is: 3, 2, 1

11. **What is the output of the below program?**

```
>>>names = ['Chris', 'Jack', 'John', 'Daman']
>>>print(names[-1][-1])
```

The output is: n

12. **What is Enumerate() Function in Python?**

The Python enumerate() function adds a counter to an iterable object. enumerate() function can accept sequential indexes starting from zero.

13. **What is data type SET in Python and how to work with it?**

The Python data type "set" is a kind of collection. It has been part of Python since version 1

14. **How do you Concatenate Strings in Python?**

We can use '+' to concatenate strings.

15. **How to generate random numbers in Python?**
    We can generate random numbers using different functions in Python. They are:

16. **How to print** sum **of the numbers starting from 1 to 100?**
    We can print sum of the numbers starting from 1 to 100 using this code:

17. **How do you set a global variable inside a function?**
    Yes, we can use a global variable in other functions by declaring it as global in each function that assigns to it:

18. **What is namespace in Python?**

    In Python, every name introduced has a place where it lives and can be hooked for. This is known as namespace. It is like a box where a variable name is mapped to the object placed. Whenever the variable is searched out, this box will be searched, to get corresponding object.

19. **What is lambda in Python?**

    It is a single expression anonymous function often used as inline function.

20. **Why lambda forms in python does not have statements?**

    A lambda form in python does not have statements as it is used to make new function object and then return them at runtime.

21. **What is pass in Python?**

    Pass means, no-operation Python statement, or in other words it is a place holder in compound statement, where there should be a blank left and nothing has to be written there.

22. **In Python what are iterators?**

    In Python, iterators are used to iterate a group of elements, containers like list.

23. **What is unittest in Python?**

    A unit testing framework in Python is known as unittest. It supports sharing of setups, automation testing, shutdown code for tests, aggregation of tests into collections etc.

24. **In Python what is slicing?**

    A mechanism to select a range of items from sequence types like list, tuple, strings etc. is known as slicing.

25. **What are generators in Python?**

    The way of implementing iterators are known as generators. It is a normal function except that it yields expression in the function.

26. **What is negative index in Python?**

    Python sequences can be index in positive and negative numbers. For positive index, 0 is the first index, 1 is the second index and so forth. For negative index, (-1) is the last index and (-2) is the second last index and so forth.

27. **How you can convert a number to a string?**

    In order to convert a number into a string, use the inbuilt function str(). If you want a octal or hexadecimal representation, use the inbuilt function oct() or hex().

28. **What is the difference between Xrange and range?**

    Xrange returns the xrange object while range returns the list, and uses the same memory and no matter what the range size is.

29. **What is module and package in Python?**

    In Python, module is the way to structure program. Each Python program file is a module, which imports other modules like objects and attributes.

30. **How can you share global variables across modules?**

To share global variables across modules within a single program, create a special module. Import the config module in all modules of your application. The module will be available as a global variable across modules.

EXPERIMENT -13

OBJECTIVE:

Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is a right triangle (Recall

from the Pythagorean Theorem that in a right triangle, the square of one side equals

the sum of the squares of the other two sides).

SOURCECODE:

```
# Python Program to find the area of triangle

a = 5
b = 6
c = 7

# Uncomment below to take inputs from the user
# a = float(input('Enter first side: '))
# b = float(input('Enter second side: '))
# c = float(input('Enter third side: '))

# calculate the semi-perimeter
s = (a + b + c) / 2

# calculate the area
area = (s*(s-a)*(s-b)*(s-c)) ** 0.5
print('The area of the triangle is %0.2f' %area)
```

INPUT ANDOUTPUT:

```
The area of the triangle is 14.70
```

VIVA Questions

Q1. How is a Python class created ?

Classes are created using the class statement. An example might be class aa rdvark(fooba r):

Q2. How is Python executed ?

Python files are compiled to bytecode. which is then executed by the host.

Alternate Answer:

Type python .pv at the command line.

Q3. What is the difference between .py and .pycfiles ?

.py files are Python source files. .pyc files are the compiled bvtecode files that is generated by the Python compiler

Q4. How do you invoke the Python interpreter for interactive use ?

python or pythonx.y where x.y are the version of the Python interpreter desired.

Q5. How are Phon blocks defined ?

By indents or tabs. This is different from most other languages which use symbols to define blocks. Indents in Python are significant.

Q6. What is the Pthon interpreter prompt ?

Three greater-than signs: >>> Also, when the int

erpreter is waiting for more input the prompt changes to three periods

Q7. How do you execute a Python Script ?

From the command line, type python .py or pythonx.y

.py where the x.y is the version of the Python interpreter desired.

Q8. Explain the use of try: except raise, and finally ?

Try, except and finally blocks are used in Python error handling. Code is executed in the try block until an error occurs. One can use a generic except block, which will receive control after all errors, or one can use specific exception handling blocks for various error types. Control is transferred to the appropriate except block. In all cases, the finally block is executed. Raise may be used to raise your own exceptions.

Q9. Illustrate the proper use of Python error handling ?

Code Example:

```
1 try:
2 ….#This can be any code
3 except:
4 …# error handling code goes here
5 finally.
6 …# code that will be executed regardless of exception handling goes here.
```

Q10. What happens if an error occurs that is not handled in the except block ?

The program tenuinates. and an execution trace is sent to sys.stderr.

Q11. How are modules used in a Python program ?

Modules are brought in via the import statement.

Q12. How do you create a Python function ?

Functions are defined using the def statement. An example might be def foo(bar):

Q13. How is a Python class created ?

Classes are created using the class statement. An example might be class aa rdvark(fooba r):

Q14. How is a Python class instantiated ?

The class is instantiated by calling it directly. An example might be

myclass =aardvark(5)

Q15. In a class definition, what does the __ init_O function do ?

It overrides the any initialization from an inherited class, and is called when the class is instantiated

Q16. How does a function return values ?

Functions return values using the return statement.

Q17. What happens when a function doesn't have a return statement ? Is this valid ?

Yes, this is valid. The function will then return a None object. The end of a function is defined by the block of code being executed (i.e., the indenting) not by any explicit keyword.

Q18. What is the lambda operator ?

The lambda operator is used to create anonymous functions. It is mostly used in cases where one wishes to pass functions as parameters. or assign them to variable names.

Q19. Explain the difference between local and global namespaces ?

Local namespaces are created within a function. when that function is called. Global name spaces are created when the program starts.

Q20. Name the four main types of namespaces in Python ?

- Global,
- Local,
- Module and
- Class namespaces.

Q21. When would you use triple quotes as a delimiter ?

Triple quotes ‘’’” or ‘“ are string delimiters that can span multiple lines in Python. Triple quotes are usually used when spanning multiple lines, or enclosing a string that has a mix of single and double quotes contained therein.

Q22. What are the two major loop statements ?

for and while

Q23. Under what circumstances would von use a while statement rather than for ?

The while statement is used for simple repetitive looping and the for statement is used when one wishes to iterate through a list of items, such as database records, characters in a string, etc.

Q24. What happens if.ou put an else statement after after block ?

The code in the else block is executed after the for loop completes, unless a break is encountered in the for loop execution. in which case the else block is not executed.

Q25. Explain the use of break and continue in Python looping ?

The break statement stops the execution of the current loop. and transfers control to the next block. The continue statement ends the current block's execution and jumps to the next iteration of the loop.

Q26. When would you use a continue statement in a for loop ?

When processing a particular item was complete; to move on to the next, without executing further processing in the block. The continue statement says, "I'm done processing this item, move on to the next item."

Q27. When would you use a break statement in a for loop ?

When the loop has served its purpose. As an example. after finding the item in a list searched for, there is no need to keep looping. The break statement says, I'm done in this loop; move on to the next block of code."

Q28. What is the structure of aforloop ?

for in : … The ellipsis represents a code block to be executed, once for each item in the sequence. Within the block, the item is available as the current item from the entire list.

Q29. What is the structure of a while loop?

while : … The ellipsis represents a code block to be executed. until the condition becomes false. The condition is an expression that is considered true unless it evaluates to o, null or false.

Q30. Use a for loop and illustrate how you would define and print the characters in a string out, one per line ?

```
1    myString = "I Love Python"
2    formyChar hi myString:
3    print myChar
```

EXPERIMENT -14

OBJECTIVE:

Write a python program to define a module to find Fibonacci Numbers and import the module to another program.

SOURCECODE:

```
# Program to display the Fibonacci sequence up to n-th term where n is provided by the user
# change this value for a different result
```

```
nterms = 10
# uncomment to take input from the user
#nterms = int(input("How many terms? "))
# first two terms
n1 = 0
n2 = 1
count = 0
# check if the number of terms is valid
if nterms<= 0:
print("Please enter a positive integer")
elifnterms == 1:
print("Fibonacci sequence upto",nterms,":")
  print(n1)
else:
print("Fibonacci sequence upto",nterms,":")
  while count <nterms:
print(n1,end=' , ')
      nth = n1 + n2
      # update values
      n1 = n2
      n2 = nth
count += 1
```

INPUT ANDOUTPUT:

```
Fibonacci sequence upto10 :

0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
```

VIVA Questions

1. Given the string “I LoveQPython” use afor loop and illustrate printing each character tip to, but not including the Q ?
   inyString = “I Love Pijtlzon”
   formyCizar inmyString:
   fmyC’har ==
   break
   print myChar

2. Given the string “I Love Python” print out each character except for the spaces, using a for loop ?
   inyString = I Love Python”
   formyCizar inmyString:
   fmyChar == ‘’ ‘’:

continue
print myChar

3. how to execute a ioop ten times ?
   i=1
   whilei< 10:
4. How to use GUI that comes with Python to test your code?
   That is just an editor and a graphical version of the interactive shell. You write or load code and run it, or type it into the shell.
   There is no automated testing.
5. What is Python good for ?
   Python is a high-level general-purpose programming language that can be applied to many different classes of problems.
6. How does the Python version numbering scheme work ?
   Python versions are numbered A.B.C or A.B.
   A is the major version number. It is only incremented for major changes in the language.
   B is the minor version number, incremented for less earth-shattering changes.
   C is the micro-level. It is incremented for each bug fix release.
7. Where is math.py (socket.py, regex.py, etc.) source file?
   If you can't find a source file for a module, it may be a built-in or dynamically loaded module implemented in C, C++ or other compiled language. In this case you may not have the source file or it may be something like mathmodule.c, somewhere in a C source directory (not on the Python Path).
8. How do I make a Python script executable on UNIX ?
   You need to do two things :
   The script file's mode must be executable and the first line must begin with "#!" followed by the path of the Python interpreter. The first is done by executing chmod +x scriptfile or perhaps chmod 755 'script' file.
   The second can be done in a number of ways.
9. Why don't my signal handlers work ?
   The most common problem is that the signal handler is declared with the wrong argument list. It is called as:
   handler (signum, frame)
   So it should be declared with two arguments:
   def handler(signum, frame):
10. How do I test a Python program or component?
    Python comes with two testing frameworks :
    The documentation test modulefinds examples in the documentation strings for a module and runs them, comparing the output with the expected output given in the documentation string.
    The unit test moduleis a fancier testing framework modeled on Java and Smalltalk testing frameworks.
11. How do I find undefined g++ symbols __builtin_new or __pure_virtual?
    To dynamically load g++ extension modules, you must :
    Recompile Python
    Re-link it using g++ (change LINKCC in the python Modules Makefile)
    Link your extension module using g++ (e.g., "g++ -shared -o mymodule.so mymodule.o").
12. How do I send mail from a Python script?
    Use the standard library module smtplib. Here's a very simple interactive mail sender that uses it. This method will work on any host that supports an SMTP listener.
13. How can I mimic CGI form submission (METHOD=POST)? I would like to retrieve web pages that are the result of posting a form. Is there existing code that would let me do this easily?
14. Why is that none of my threads are not running? How can I make it work?
    As soon as the main thread exits, all threads are killed. Your main thread is running too quickly, giving the threads no time to do any work.
    A simple fix is to add a sleep to the end of the program that's long enough for all the threads to finish:
    import threading, time
    def thread_task(name, n):

fori inrange(n): print name, i
fori inrange(10)

15. Installation of Python 3.6.1 ?
    Download the required 3.6.1 python, executable installer file from the www.python.org.com website.
16. What Are The Implementation In Python Program?
    Python program can be implemented by two ways
    Interactive Mode (Submit statement by statement explicitly)
    2. Batch Mode (Writing all statements and submit all statements)
    In Interactive mode python command shell is required. It is available in installation of python cell.
    17. What Are The Types of Objects Support in Python Language?
    Immutable Objects
    Mutable Objects
17. Q49. Control flow statements?
    By default, python program execution starts from the first line, execute each and every statement only once and transactions the program if the last statement of the program execution is over.
    Control flow statements are used to disturb the normal flow of the execution of the program.
18. What is a Tuple?
    Tuple Objects can be created by using parenthesis or by calling tuple function or by assigning multiple values to a single variable
    Tuple objects are immutable objects
    Incision order is preserved
    Duplicate elements are allowed
19. What is the difference Between List And Tuple?

List	Tuple
List objects are mutable objects	Tuple objects are immutable Objects
Applying iterations on list objects takes longer time	Applying iterations on tuple Objects takes less time
If the frequent operation is the insertion or deletion of the elements then it is recommended to use a list	If the frequent operation is the retrieval of the elements then it is recommended to use a tuple
List can't be used as a 'key' for the dictionary	Tuple can be used as a key for the dictionary if the tuple is storing only immutable elements

20. What is Anonymous Function or Lambda Function?
    A function which doesn't contain any name is known as an anonymous function lambda function
21. Modules Search Path?
    By default python interpreter search for the imported modules in the following locations:
22. What are the Packages?
    A package is nothing but a folder or dictionary which represents a collection of modules
    A package can also contain sub packages
    We can import the modules of the package by using package name.module name
    We can import the modules of the package by using package name.subpackagename.module name
23. What is File Handling?
    File is a named location on the disk, which stores the data in permanent manner.

24. What are the Runtime Errors?

The errors which occur after starting the execution of the programs are known as runtime errors.

25. What is Abnormal Termination?

The concept of terminating the program in the middle of its execution without executing the last statement of the main module is known as an abnormal termination

Abnormal termination is an undesirable situation in programming languages.

26. What is Try Block?

A block which is preceded by the try keyword is known as a try block

27. What is the Encapsulation?

The concept of binding or grouping related data members along with its related functionalities is known as an Encapsulation.

28. . What is Garbage Collection?

The concept of removing unused or unreferenced objects from the memory location is known as a Garbage Collection.

29. Executing DML Commands Through Python Programs?

DML Commands are used to modify the data of the database objects

30. What is Multithreading?

Thread Is a functionality or logic which can execute simultaneously along with the other part of the program

EXPERIMENT -15

OBJECTIVE:

Write a python program to define a module and import a specific function in that module to another program.

SOURCECODE:

```
# Python Program to find numbers divisible by thirteen from a list using anonymous function

# Take a list of numbers
my_list = [12, 65, 54, 39, 102, 339, 221,]

# use anonymous function to filter
result = list(filter(lambda x: (x % 13 == 0), my_list))

# display the result
print("Numbers divisible by 13 are",result)
```

INPUT ANDOUTPUT:

```
Numbers divisible by 13 are [65, 39, 221]
```

VIVA Questions

1) What is Python? What are the benefits of using Python?
Python is a programming language with objects, modules, threads, exceptions and automatic memory management. The benefits of pythons are that it is simple and easy, portable, extensible, build-in data structure and it is an open source.

2) What is PEP 8?
PEP 8 is a coding convention, a set of recommendation, about how to write your Python code more readable.

3) What is pickling and unpickling?
Pickle module accepts any Python object and converts it into a string representation and dumps it into a file by using dump function, this process is called pickling. While the process of retrieving original Python objects from the stored string representation is called unpickling.

4) How Python is interpreted?
Python language is an interpreted language. Python program runs directly from the source code. It converts the source code that is written by the programmer into an intermediate language, which is again translated into machine language that has to be executed.

5) How memory is managed in Python?
Python memory is managed by Python private heap space. All Python objects and data structures are located in a private heap. The programmer does not have an access to this private heap and interpreter takes care of this Python private heap. The allocation of Python heap space for Python objects is done by Python memory manager. The core API gives access to some tools for the programmer to code.
Python also have an inbuilt garbage collector, which recycle all the unused memory and frees the memory and makes it available to the heap space.
python

6) What are the tools that help to find bugs or perform static analysis?
PyChecker is a static analysis tool that detects the bugs in Python source code and warns about the style and complexity of the bug. Pylint is another tool that verifies whether the module meets the coding standard.

7) What are Python decorators?
A Python decorator is a specific change that we make in Python syntax to alter functions easily.

8) What is the difference between list and tuple?
The difference between list and tuple is that list is mutable while tuple is not. Tuple can be hashed for e.g as a key for dictionaries.

9) How are arguments passed by value or by reference?
Everything in Python is an object and all variables hold references to the objects. The references values are according to the functions; as a result you cannot change the value of the references. However, you can change the objects if it is mutable.

10) What is Dict and List comprehensions are?
They are syntax constructions to ease the creation of a Dictionary or List based on existing iterable.

11) What are the built-in type does python provides?
There are mutable and Immutable types of Pythons built in types Mutable built-in types
List
Sets
Dictionaries
Immutable built-in types
Strings
Tuples
Numbers

12) What is namespace in Python?
In Python, every name introduced has a place where it lives and can be hooked for. This is known as namespace. It is like a box where a variable name is mapped to the object placed. Whenever the variable is searched out, this box will be searched, to get corresponding object.

13) What is lambda in Python?
It is a single expression anonymous function often used as inline function.

14) Why lambda forms in python does not have statements?
A lambda form in python does not have statements as it is used to make new function object and then return them at runtime.

15) What is pass in Python?
Pass means, no-operation Python statement, or in other words it is a place holder in compound statement, where there should be a blank left and nothing has to be written there.

16) In Python what are iterators?
In Python, iterators are used to iterate a group of elements, containers like list.

17) What is unittest in Python?
A unit testing framework in Python is known as unittest. It supports sharing of setups, automation testing, shutdown code for tests, aggregation of tests into collections etc.

18) In Python what is slicing?
A mechanism to select a range of items from sequence types like list, tuple, strings etc. is known as slicing.

19) What are generators in Python?
The way of implementing iterators are known as generators. It is a normal function except that it yields expression in the function.

20) What is docstring in Python?
A Python documentation string is known as docstring, it is a way of documenting Python functions, modules and classes.

21) How can you copy an object in Python?
To copy an object in Python, you can try copy.copy () or copy.deepcopy() for the general case. You cannot copy all objects but most of them.

22) What is negative index in Python?
Python sequences can be index in positive and negative numbers. For positive index, 0 is the first index, 1 is the second index and so forth. For negative index, (-1) is the last index and (-2) is the second last index and so forth.

23) How you can convert a number to a string?
In order to convert a number into a string, use the inbuilt function str(). If you want a octal or hexadecimal representation, use the inbuilt function oct() or hex().

24) What is the difference between Xrange and range?
Xrange returns the xrange object while range returns the list, and uses the same memory and no matter what the range size is.

25) What is module and package in Python?
In Python, module is the way to structure program. Each Python program file is a module, which imports other modules like objects and attributes.
The folder of Python program is a package of modules. A package can have modules or subfolders.

26) Mention what are the rules for local and global variables in Python?
Local variables: If a variable is assigned a new value anywhere within the function's body, it's assumed to be local.
Global variables: Those variables that are only referenced inside a function are implicitly global.

27) How can you share global variables across modules?
To share global variables across modules within a single program, create a special module. Import the config module in all modules of your application. The module will be available as a global variable across modules.

28) Explain how can you make a Python Script executable on Unix?
To make a Python Script executable on Unix, you need to do two things,
Script file's mode must be executable and
the first line must begin with # (#!/usr/local/bin/python)

29) Explain how to delete a file in Python?
By using a command os.remove (filename) or os.unlink(filename)

30) Explain how can you generate random numbers in Python?
To generate random numbers in Python, you need to import command as
import random
random.random()
This returns a random floating point number in the range [0,1)

EXPERIMENT -16

OBJECTIVE:

Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.

SOURCECODE:

```
withopen("test.txt")as f:
withopen("out.txt","w")as f1:
for line in f:
            f1.write(line)
```

INPUT ANDOUTPUT:

```
Case 1:
Contents of file(test.txt):
Hello world

Output(out.text):
Hello world

Case 2:
Contents of file(test.txt):
Sanfoundry

Output(out.text):
Sanfoundry
```

VIVA Questions

1. What is the output when following statement is executed ?

```
1. >>>"a"+"bc"
```

abc

2. What is the output when following statement is executed ?

```
>>>"abcd"[2:]
```

cd

3. The output of executing string.ascii_letters can also be achieved by:

string.ascii_lowercase+string.ascii_upercase

4. What is the output when following code is executed ?

```
>>> str1 = 'hello'
>>> str2 = ','
>>> str3 = 'world'
>>> str1[-1:]
```

o

5. What arithmetic operators cannot be used with strings ?

–

6. What is the output when following code is executed ?
advertisement

```
>>>print (r"\nhello")
```

\nhello

7. What is the output when following statement is executed ?

```
>>>print('new' 'line')
```

newline

8. What is the output when following statement is executed ?

```
>>> print('x\97\x98')
```

x\97

9. What is the output when following code is executed ?

```
1. >>>str1="helloworld"
2. >>>str1[::-1]
```

dlrowolleh

10. print(0xA + 0xB + 0xC) :
33
View Answer
11. What is the output when following statement is executed ?

```
1. >>>chr(ord('A'))
```

a) A
b) B
c) a
d) Error
View Answer
Answer: a
Explanation: Execute in shell to verify.
12. What is the output when following statement is executed ?

```
1. >>>print(chr(ord('b')+1))
```

a) a
b) b
c) c
d) A
View Answer
Answer: c
Explanation: Execute in the shell to verify.
13. Which of the following statement prints hello\example\test.txt ?
a) print("hello\example\test.txt")
b) print("hello\\example\\test.txt")

c) print(“hello\”example\”test.txt”)
d) print(“hello”\example”\test.txt”)
View Answer
Answer: b
Explanation: \is used to indicate that the next \ is not an escape sequence.
14. Suppose s is “\t\tWorld\n”, what is s.strip() ?
a) \t\tWorld\n
b) \t\tWorld\n
c) \t\tWORLD\n
d) World
View Answer
Answer: d
Explanation: Execute help(string.strip) to find details.
15. The format function, when applied on a string returns :
a) Error
b) int
c) bool
d) str
View Answer
Answer: d
Explanation: Format function returns a string.
16. What is the output of “hello”+1+2+3 ?
a) hello123
b) hello
c) Error
d) hello6
View Answer
Answer: c
Explanation: Cannot concatenate str and int objects.
17. What is the output when following code is executed ?

```
>>>print("D", end = ' ')
>>>print("C", end = ' ')
>>>print("B", end = ' ')
>>>print("A", end = ' ')
```

a) DCBA
b) A, B, C, D
c) D C B A
d) D, C, B, A will be displayed on four lines
View Answer
Answer: d
Explanation: Execute in the shell.

18. What is the output when following statement is executed ?(python 3.xx)

```
>>>print(format("Welcome", "10s"), end = '#')
>>>print(format(111, "4d"), end = '#')
>>>print(format(924.656, "3.2f"))
```

a) Welcome# 111#924.66
b) Welcome#111#924.66
c) Welcome#111#.66
d) Welcome # 111#924.66
View Answer
Answer: d
Explanation: Execute in the shell to verify.
19. What will be displayed by print(ord(‘b’) – ord(‘a’)) ?
a) 0
b) 1
c) -1
d) 2
View Answer

Answer: b
Explanation: ASCII value of b is one more than a. Hence the output of this code is 98-97, which is equal to 1.
20. Say s="hello" what will be the return value of type(s) ?
a) int
b) bool
c) str
d) String
View Answer
Answer: c
Explanation: str is used to represent strings in python.

21. What is "Hello".replace("l", "e")
a) Heeeo
b) Heelo
c) Heleo
d) None
View Answer
Answer: a
Explanation: Execute in shell to verify.
22. To retrieve the character at index 3 from string s="Hello" what command do we execute (multiple answers allowed) ?
a) s[].
b) s.getitem(3)
c) s.__getitem__(3)
d) s.getItem(3)
View Answer
Answer: c
Explanation: __getitem(..) can be used to get character at index specified as parameter.
23. To return the length of string s what command do we execute ?
a) s.__len__()
b) len(s)
c) size(s)
d) s.size()
View Answer
Answer: a
Explanation: Execute in shell to verify.
24. If a class defines the __str__(self) method, for an object obj for the class, you can use which command to invoke the __str__ method.
a) obj.__str__()
b) str(obj)
c) print obj
d) All of the mentioned
View Answer
Answer: d
Explanation: Execute in shell to verify.
25. To check whether string s1 contains another string s2, use
a) s1.__contains__(s2)
b) s2 in s1
c) s1.contains(s2)
d) si.in(s2)
View Answer
Answer: a
Explanation: s2 in s1 works in the same way as calling the special function __contains__ .
26. Suppose i is 5 and j is 4, i + j is same as
a) i.__add(j)
b) i.__add__(j)
c) i.__Add(j)
d) i.__ADD(j)
View Answer

Answer: b
Explanation: Execute in shell to verify.
27. What is the output of the following code ?

```
1. class Count:
2.     def __init__(self, count = 0):
3. self.__count = count
4.
5. c1 = Count(2)
6. c2 = Count(2)
7. print(id(c1) == id(c2), end = " ")
8.
9. s1 = "Good"
10.  s2 = "Good"
11.  print(id(s1) == id(s2))
```

a) True False
b) True True
c) False True
d) False False
View Answer
Answer: c
Explanation: Execute in the shell objects cannot have same id, however in the case of strings its different.
28. What is the output of the following code ?

```
1. class Name:
2.     def __init__(self, firstName, mi, lastName):
3. self.firstName = firstName
4. self.mi = mi
5. self.lastName = lastName
6.
7. firstName = "John"
8. name = Name(firstName, 'F', "Smith")
9. firstName = "Peter"
10.  name.lastName = "Pan"
11.  print(name.firstName, name.lastName)
```

a) Peter Pan
b) John Pan
c) Peter Smith
d) John Smith
View Answer
Answer: b
Explanation: Execute in the shell to verify.

29. What function do you use to read a string?
a) input(“Enter a string”)
b) eval(input(“Enter a string”))
c) enter(“Enter a string”)
d) eval(enter(“Enter a string”))
View Answer
Answer: a
Explanation: Execute in shell to verify.
30. Suppose x is 345.3546, what is format(x, “10.3f”) (_ indicates space)
a) __345.355
b) ___345.355
c) ____345.355
d) _____345.354
View Answer
Answer: b
Explanation: Execute in the shell to verify.

EXPERIMENT -17

OBJECTIVE:

Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.

SOURCECODE:

```
    items = input("Input comma separated sequence of words")
    words = [word for word in items.split(",")]
print(",".join(sorted(list(set(words)))))
```

INPUT ANDOUTPUT:

```
Input comma separated sequence of words red, black, pink,
green
 black, green, pink, red
```

VIVA Questions

1. What is the output of the following?

```
print("xyyzxyzxzxyy".count('yy'))
```

a) 2

b) 0

c) error

d) none of the mentioned

View Answer
Answer: a
Explanation: Counts the number of times the substring 'yy' is present in the given string.

2. What is the output of the following?

```
print("xyyzxyzxzxyy".count('yy',1))
```

a) 2

b) 0

c) 1

d) none of the mentioned

View Answer
Answer: a
Explanation: Counts the number of times the substring 'yy' is present in the given string, starting from position 1.

3. What is the output of the following?
```
print("xyyzxyzxzxyy".count('yy',2))
```

a) 2

b) 0

c) 1

d) none of the mentioned

View Answer
Answer: c
Explanation: Counts the number of times the substring 'yy' is present in the given string, starting from position 2.

4. What is the output of the following?
```
print("xyyzxyzxzxyy".count('xyy',0,100))
```

a) 2

b) 0

c) 1

d) error

View Answer
Answer: a
Explanation: An error will not occur if the end value is greater than the length of the string itself.

5. What is the output of the following?
```
print("xyyzxyzxzxyy".count('xyy',2,11))
```

a) 2

b) 0

c) 1

d) error

View Answer
Answer: b
Explanation: Counts the number of times the substring 'xyy' is present in the given string, starting from position 2 and ending at position 11.

6. What is the output of the following?
```
print("xyyzxyzxzxyy".count('xyy', -10, -1))
```

a) 2

b) 0

c) 1

d) error

View Answer
Answer: b
Explanation: Counts the number of times the substring 'xyy' is present in the given string, starting from position 2 and ending at position 11.
advertisement

7. What is the output of the following?
```
print('abc'.encode())
```

a) abc

b) 'abc'

c) b'abc'

d) h'abc'

View Answer
Answer: c
Explanation: A bytes object is returned by encode.

8. What is the default value of encoding in encode()?

a) ascii

b) qwerty

c) utf-8

d) utf-16

View Answer
Answer: c
Explanation: The default value of encoding is utf-8.

9. What is the output of the following?
```
print("xyyzxyzxzxyy".endswith("xyy"))
```

a) 1

b) True

c) 3

d) 2

View Answer
Answer: b
Explanation: The function returns True if the given string ends with the specified substring.

10. What is the output of the following?
```
print("xyyzxyzxzxyy".endswith("xyy",0,2))
```

a) 0

b) 1

c) True

d) False

View Answer

Answer: d
Explanation: The function returns False if the given string does not end with the specified substring.

11. What is the output of the following?
print('{0:.2}'.format(1/3))

a) 0.333333

b) 0.33

c) 0.333333:.2

d) Error

View Answer
Answer: b
Explanation: .2 specifies the precision.

12. What is the output of the following?
print('{0:.2%}'.format(1/3))

a) 0.33

b) 0.33%

c) 33.33%

d) 33%

View Answer
Answer: c
Explanation: The symbol % is used to represent the result of an expression as a percentage.

13. What is the output of the following?
print('ab12'.isalnum())

a) True

b) False

c) None

d) Error

View Answer
Answer: a
Explanation: The string has only letters and digits.

14. What is the output of the following?
print('ab,12'.isalnum())

a) True

b) False

c) None

d) Error

View Answer
Answer: b
Explanation: The character , is not a letter or a digit.

15. What is the output of the following?
print('ab'.isalpha())

a) True

b) False

c) None

d) Error

View Answer
Answer: a
Explanation: The string has only letters.

16. What is the output of the following?
print('a B'.isalpha())

a) True

b) False

c) None

d) Error

View Answer
Answer: b
Explanation: Space is not a letter.

17. What is the output of the following?
print('0xa'.isdigit())

a) True

b) False

c) None

d) Error

View Answer
Answer: b
Explanation: Hexadecimal digits aren’t considered as digits (a-f).

18. What is the output of the following?
print(''.isdigit())

a) True

b) False

c) None

d) Error

View Answer
Answer: b
Explanation: If there are no characters then False is returned.

19. What is the output of the following?
print('my_string'.isidentifier())

a) True

b) False

c) None

d) Error

View Answer
Answer: a
Explanation: It is a valid identifier.

20. What is the output of the following?
```
print('__foo__'.isidentifier())
```

a) True

b) False

c) None

d) Error

View Answer
Answer: a
Explanation: It is a valid identifier.

21. What is the output of the following?
```
print('for'.isidentifier())
```

a) True

b) False

c) None

d) Error

View Answer
Answer: a
Explanation: Even keywords are considered as valid identifiers.

22. What is the output of the following?
```
print('abc'.islower())
```

a) True

b) False

c) None

d) Error

View Answer
Answer: a
Explanation: There are no uppercase letters.

23. What is the output of the following?
```
print('a@ 1,'.islower())
```

a) True

b) False

c) None

d) Error

View Answer
Answer: a
Explanation: There are no uppercase letters.

24. What is the output of the following?
```
print('11'.isnumeric())
```

a) True

b) False

c) None

d) Error

View Answer
Answer: a
Explanation: All the character are numeric.

25. What is the output of the following?
```
print('1.1'.isnumeric())
```

a) True

b) False

c) None

d) Error

View Answer
Answer: b
Explanation: The character . is not a numeric character.

26. What is the output of the following?
```
print('1@ a'.isprintable())
```

a) True

b) False

c) None

d) Error

View Answer
Answer: a
Explanation: All those characters are printable.

27. What is the output of the following?
```
print("""".isspace())
```

a) True

b) False

c) None

d) Error

View Answer
Answer: b
Explanation: None.

28. What is the output of the following?
```
print('\t'.isspace())
```

a) True

b) False

c) None

d) Error

View Answer
Answer: a
Explanation: Tab Spaces are considered as spaces.

29. What is the output of the following?
```
print('HelloWorld'.istitle())
```

a) True

b) False

c) None

d) Error

View Answer
Answer: b
Explanation: The letter W is uppercased.

30. What is the output of the following?

```
print('Hello World'.istitle())
```

a) True

b) False

c) None

d) Error

View Answer
Answer: a
Explanation: It is in title form.

EXPERIMENT -18

OBJECTIVE:

Write a Python class to convert an integer to a roman numeral.

SOURCECODE:

```
class py_solution:
   def int_to_Roman(self, num):
val = [
          1000, 900, 500, 400,
          100, 90, 50, 40,
          10, 9, 5, 4,
          1
          ]
syb = [
          "M", "CM", "D", "CD",
          "C", "XC", "L", "XL",
          "X", "IX", "V", "IV",
          "I"
          ]
roman_num = ''
i = 0
while  num> 0:
          for _ in range(num // val[i]):
roman_num += syb[i]
              num -= val[i]
i += 1
      return roman_num
print(py_solution().int_to_Roman(1))
print(py_solution().int_to_Roman(4000))
```

INPUT ANDOUTPUT:

```
I
MMMM
```

VIVA Questions

1. Which of the following commands will create a list?

a) list1 = list()

b) list1 = [].

c) list1 = list([1, 2, 3])

d) all of the mentioned

View Answer
Answer: d
Explanation: Execute in the shell to verify

2. What is the output when we execute list("hello")?

a) ['h', 'e', 'l', 'l', 'o'].

b) ['hello'].

c) ['llo'].

d) ['olleh'].

View Answer
Answer: a
Explanation: Execute in the shell to verify.

3. Suppose listExample is ['h','e','l','l','o'], what is len(listExample)?

a) 5

b) 4

c) None

d) Error

View Answer
Answer: a
Explanation: Execute in the shell and verify.

4. Suppose list1 is [2445,133,12454,123], what is max(list1) ?

a) 2445

b) 133

c) 12454

d) 123

View Answer
Answer: c
Explanation: Max returns the maximum element in the list.

5. Suppose list1 is [3, 5, 25, 1, 3], what is min(list1) ?

a) 3

b) 5

c) 25

d) 1

View Answer
Answer: d
Explanation: Min returns the minimum element in the list.

6. Suppose list1 is [1, 5, 9], what is sum(list1) ?

a) 1

b) 9

c) 15

d) Error

View Answer
Answer: c
Explanation: Sum returns the sum of all elements in the list.

7. To shuffle the list(say list1) what function do we use ?

a) list1.shuffle()

b) shuffle(list1)

c) random.shuffle(list1)

d) random.shuffleList(list1)

View Answer
Answer: c
Explanation: Execute in the shell to verify .

8. Suppose list1 is [4, 2, 2, 4, 5, 2, 1, 0], Which of the following is correct syntax for slicing operation ?

a) print(list1[0])

b) print(list1[:2])

c) print(list1[:-2])

d) all of the mentioned

View Answer
Answer: d
Explanation: Slicing is allowed in lists just as in the case of strings.

9. Suppose list1 is [2, 33, 222, 14, 25], What is list1[-1] ?

a) Error

b) None

c) 25

d) 2

View Answer
Answer: c
Explanation: -1 corresponds to the last index in the list.

10. Suppose list1 is [2, 33, 222, 14, 25], What is list1[:-1] ?

a) [2, 33, 222, 14].

b) Error

c) 25

d) [25, 14, 222, 33, 2].

View Answer
Answer: a
Explanation: Execute in the shell to verify.

11. Suppose list1 is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after list1.reverse() ?

a) [3, 4, 5, 20, 5, 25, 1, 3].

b) [1, 3, 3, 4, 5, 5, 20, 25].

c) [25, 20, 5, 5, 4, 3, 3, 1].

d) [3, 1, 25, 5, 20, 5, 4, 3].

View Answer
Answer: d
Explanation: Execute in the shell to verify.

12. Suppose listExample is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after listExample.extend([34, 5]) ?

a) [3, 4, 5, 20, 5, 25, 1, 3, 34, 5].

b) [1, 3, 3, 4, 5, 5, 20, 25, 34, 5].

c) [25, 20, 5, 5, 4, 3, 3, 1, 34, 5].

d) [1, 3, 4, 5, 20, 5, 25, 3, 34, 5].

View Answer
Answer: a
Explanation: Execute in the shell to verify.

13. Suppose listExample is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after listExample.pop(1) ?

a) [3, 4, 5, 20, 5, 25, 1, 3].

b) [1, 3, 3, 4, 5, 5, 20, 25].

c) [3, 5, 20, 5, 25, 1, 3].

d) [1, 3, 4, 5, 20, 5, 25].

View Answer
Answer: c
Explanation: pop() removes the element at the position specified in the parameter.

14. Suppose listExample is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after listExample.pop()?

a) [3, 4, 5, 20, 5, 25, 1].

b) [1, 3, 3, 4, 5, 5, 20, 25].

c) [3, 5, 20, 5, 25, 1, 3].

d) [1, 3, 4, 5, 20, 5, 25].

View Answer
Answer: a
Explanation: pop() by default will remove the last element.

15. What is the output when the following code is executed ?

```
1. >>>"Welcome to Python".split()
```

a) [“Welcome”, “to”, “Python”].

b) (“Welcome”, “to”, “Python”)

c) {“Welcome”, “to”, “Python”}

d) “Welcome”, “to”, “Python”

View Answer
Answer: a
Explanation: split() function returns the elements in a list.

16. What is the output when following code is executed ?

```
1. >>>list("a#b#c#d".split('#'))
```

a) [‘a’, ‘b’, ‘c’, ‘d’].

b) [‘a b c d’].

c) [‘a#b#c#d’].

d) [‘abcd’].

View Answer
Answer: a
Explanation: Execute in the shell to verify.

17. What is the output when following code is executed ?

```
1. myList = [1, 5, 5, 5, 5, 1]
2. max = myList[0]
3. indexOfMax = 0
4. for i in range(1, len(myList)):
5.     if myList[i] > max:
6.         max = myList[i]
7. indexOfMax = i
8.
9. >>>print(indexOfMax)
```

a) 1

b) 2

c) 3

d) 4

View Answer
Answer: a
Explanation: First time the highest number is encountered is at index 1.

18. What is the output when following code is executed ?

```
1. myList = [1, 2, 3, 4, 5, 6]
2. for i in range(1, 6):
3. myList[i - 1] = myList[i]
4.
5. for i in range(0, 6):
6.     print(myList[i], end = " ")
```

a) 2 3 4 5 6 1

b) 6 1 2 3 4 5

c) 2 3 4 5 6 6

d) 1 1 2 3 4 5

View Answer
Answer: c
Explanation: Execute in the shell to verify.

19. What is the output when following code is executed ?

```
1. >>>list1 = [1, 3]
2. >>>list2 = list1
3. >>>list1[0] = 4
4. >>>print(list2)
```

a) [1, 3].

b) [4, 3].

c) [1, 4].

d) [1, 3, 4].

View Answer
Answer: b
Explanation: Lists should be copied by executing [:] operation.

20. What is the output when following code is executed ?

```
1. def f(values):
2. values[0] = 44
3.
4. v = [1, 2, 3]
5. f(v)
6. print(v)
```

a) [1, 44].

b) [1, 2, 3, 44].

c) [44, 2, 3].

d) [1, 2, 3].

View Answer
Answer: c
Explanation: Execute in the shell to verify.

21. Which of the following is a Python tuple?

a) [1, 2, 3].

b) (1, 2, 3)

c) {1, 2, 3}

d) {}

View Answer
Answer: b
Explanation: Tuples are represented with round brackets.

22. Suppose t = (1, 2, 4, 3), which of the following is incorrect?

a) print(t[3])

b) t[3] = 45

c) print(max(t))

d) print(len(t))

View Answer
Answer: b
Explanation: Values cannot be modified in the case of tuple, that is, tuple is immutable.

23. What will be the output?

```
1. >>>t=(1,2,4,3)
2. >>>t[1:3]
```

a) (1, 2)

b) (1, 2, 4)

c) (2, 4)

d) (2, 4, 3)

View Answer
Answer: c
Explanation: Slicing in tuples takes place just as it does in strings.

24. What will be the output?

```
1. >>>t=(1,2,4,3)
2. >>>t[1:-1]
```

a) (1, 2)

b) (1, 2, 4)

c) (2, 4)

d) (2, 4, 3)

View Answer
Answer: c
Explanation: Slicing in tuples takes place just as it does in strings.

25. What will be the output?

```
1. >>>t = (1, 2, 4, 3, 8, 9)
2. >>>[t[i] for i in range(0, len(t), 2)]
```

a) [2, 3, 9].

b) [1, 2, 4, 3, 8, 9].

c) [1, 4, 8].

d) (1, 4, 8)

View Answer
Answer: c
Explanation: Execute in the shell to verify.

26. What will be the output?

```
1. d = {"john":40, "peter":45}
2. d["john"]
```

a) 40

b) 45

c) “john”

d) “peter”

View Answer
Answer: a
Explanation: Execute in the shell to verify.

27. What will be the output?

```
1. >>>t = (1, 2)
2. >>>2 * t
```

a) (1, 2, 1, 2)

b) [1, 2, 1, 2].

c) (1, 1, 2, 2)

d) [1, 1, 2, 2].

View Answer
Answer: a
Explanation: * operator concatenates tuple.
advertisement

28. What will be the output?

```
1. >>>t1 = (1, 2, 4, 3)
2. >>>t2 = (1, 2, 3, 4)
3. >>>t1 < t2
```

a) True

b) False

c) Error

d) None

View Answer
Answer: b
Explanation: Elements are compared one by one in this case.

29. What will be the output?

```
1. >>>my_tuple = (1, 2, 3, 4)
2. >>>my_tuple.append( (5, 6, 7) )
3. >>>print len(my_tuple)
```

a) 1

b) 2

c) 5

d) Error

View Answer
Answer: d
Explanation: Tuples are immutable and don't have an append method. An exception is thrown in this case.

30. What will be the output?

```
1. numberGames = {}
2. numberGames[(1,2,4)] = 8
3. numberGames[(4,2,1)] = 10
4. numberGames[(1,2)] = 12
5. sum = 0
6. for k in numberGames:
7.     sum += numberGames[k]
8. print len(numberGames) + sum
```

a) 30

b) 24

c) 33

d) 12

View Answer

Answer: c
Explanation: Tuples can be used for keys into dictionary. The tuples can have mixed length and the order of the items in the tuple is considered when comparing the equality of the keys.

EXPERIMENT -19

OBJECTIVE:

Write a Python class to implement pow(x, n)

SOURCECODE:

```
class py_solution:
  def pow(self, x, n):
      if x==0 or x==1 or n==1:
         return x
      if x==-1:
         if n%2 ==0:
            return 1
         else:
            return -1
      if n==0:
         return 1
      if n<0:
         return 1/self.pow(x,-n)
   val = self.pow(x,n//2)
      if n%2 ==0:
         return val*val
      return val*val*x
   print(py_solution().pow(2, -3));
   print(py_solution().pow(3, 5));
print(py_solution().pow(100, 0));
```

INPUT ANDOUTPUT:

0.125

243

1

VIVA Questions

1. What is the data type of (1)?

a) Tuple

b) Integer

c) List

d) Both tuple and integer

View Answer

Answer: b

Explanation: A tuple of one element must be created as (1,).

2. If a=(1,2,3,4), a[1:-1] is

a) Error, tuple slicing doesn't exist

b) [2,3].

c) (2,3,4)

d) (2,3)

View Answer

Answer: d

Explanation: Tuple slicing exists and a[1:-1] returns (2,3).

3. What is the output of the following code?

```
>>> a=(1,2,(4,5))
>>> b=(1,2,(3,4))
>>> a<b
```

a) False

b) True

c) Error, < operator is not valid for tuples

d) Error, < operator is valid for tuples but not if there are sub-tuples

View Answer

Answer: a

Explanation: Since the first element in the sub-tuple of a is larger that the first element in the subtuple of b, False is printed.

4. What is the output of the following piece of code when executed in Python shell?

```
>>> a=("Check")*3
>>> a
```

a) ('Check','Check','Check')

b) * Operator not valid for tuples

c) ('CheckCheckCheck')

d) Syntax error

View Answer

Answer: c

Explanation: Here ("Check") is a string not a tuple because there is no comma after the element.

5. What is the output of the following code?

```
>>> a=(1,2,3,4)
>>>del(a[2])
```

a) Now, a=(1,2,4)

b) Now, a=(1,3,4)

c) Now a=(3,4)

d) Error as tuple is immutable

View Answer

Answer: d

Explanation: ‘tuple’ object doesn’t support item deletion.

6. What is the output of the following code?

```
>>> a=(2,3,4)
>>>sum(a,3)
```

a) Too many arguments for sum() method

b) The method sum() doesn’t exist for tuples

c) 12

d) 9

View Answer

Answer: c

Explanation: In the above case, 3 is the starting value to which the sum of the tuple is added to.

7. Is the following piece of code valid?

```
>>> a=(1,2,3,4)
>>>del a
```

a) No because tuple is immutable

b) Yes, first element in the tuple is deleted

c) Yes, the entire tuple is deleted

d) No, invalid syntax for del method

View Answer

Answer: c

Explanation: The command del a deletes the entire tuple.

8. What type of data is: a=[(1,1),(2,4),(3,9)]?

a) Array of tuples

b) List of tuples

c) Tuples of lists

d) Invalid type

View Answer

Answer: b

Explanation: The variable a has tuples enclosed in a list making it a list of tuples.

9. What is the output of the following piece of code?

```
>>> a=(0,1,2,3,4)
>>> b=slice(0,2)
>>> a[b]
```

a) Invalid syntax for slicing

b) [0,2].

c) (0,1)

d) (0,2)

View Answer

Answer: c

Explanation: The method illustrated in the above piece of code is that of naming of slices.

10. Is the following piece of code valid?

```
>>> a=(1,2,3)
>>> b=('A','B','C')
>>> c=zip(a,b)
```

a) Yes, c will be ((1,2,3),('A','B','C'))

b) Yes, c will be ((1,2,3),('A','B','C'))

c) No because tuples are immutable

d) No because the syntax for zip function isn't valid

View Answer

Answer: a

Explanation: Zip function combines individual elements of two iterables into tuples. Execute in Python shell to verify.

11. Which of these about a set is not true?

a) Mutable data type

b) Allows duplicate values

c) Data type with unordered values

d) Immutable data type

View Answer

Answer: d

Explanation: A set is a mutable data type with non-duplicate, unordered values, providing the usual mathematical set operations.

12. Which of the following is not the correct syntax for creating a set?

a) set([[1,2],[3,4]])

b) set([1,2,2,3,4])

c) set((1,2,3,4))

d) {1,2,3,4}

View Answer

Answer: a

Explanation: The argument given for the set must be an iterable.

13. What is the output of the following code?

```
nums=set([1,1,2,3,3,3,4,4])
print(len(nums))
```

a) 7

b) Error, invalid syntax for formation of set

c) 4

d) 8

View Answer

Answer: c

Explanation: A set doesn't have duplicate items.

14. What is the output of the following piece of code?

```
a =[5,5,6,7,7,7]
b =set(a)
def test(lst):
if lst in b:
return 1
else:
return 0
for i in filter(test, a):
print(i,end=" ")
```

a) 5 5 6

b) 5 6 7

c) 5 5 6 7 7 7

d) 5 6 7 7 7

View Answer

Answer: c

Explanation: The filter function will return all the values from list a which are true when passed to function test. Since all the members of the set are non-duplicate members of the list, all of the values will return true. Hence all the values in the list are printed.

15. Which of the following statements is used to create an empty set?

a) { }

b) set()

c) [ ].

d) ( )

View Answer

Answer: b

Explanation: { } creates a dictionary not a set. Only set() creates an empty set.

16. What is the output of the following piece of code when executed in the python shell?

```
>>> a={5,4}
>>> b={1,2,4,5}
>>> a<b
```

a) {1,2}

b) True

c) False

d) Invalid operation

View Answer

Answer: b

Explanation: a<b returns True if a is a proper subset of b.

17. If a={5,6,7,8}, which of the following statements is false?

a) print(len(a))

b) print(min(a))

c) a.remove(5)

d) a[2]=45

View Answer

Answer: d

Explanation: The members of a set can be accessed by their index values since the elements of the set are unordered.

18. If a={5,6,7}, what happens when a.add(5) is executed?

a) a={5,5,6,7}

b) a={5,6,7}

c) Error as there is no add function for set data type

d) Error as 5 already exists in the set

View Answer

Answer: b

Explanation: There exists add method for set data type. However 5 isn't added again as set consists of only non-duplicate elements and 5 already exists in the set. Execute in python shell to verify.

19. What is the output of the following code?

```
>>> a={4,5,6}
>>> b={2,8,6}
>>>a+b
```

a) {4,5,6,2,8}

b) {4,5,6,2,8,6}

c) Error as unsupported operand type for sets

d) Error as the duplicate item 6 is present in both sets

View Answer

Answer: c

Explanation: Execute in python shell to verify.

20. What is the output of the following code?

```
>>> a={4,5,6}
>>> b={2,8,6}
>>> a-b
```

a) {4,5}

b) {6}

c) Error as unsupported operand type for set data type

d) Error as the duplicate item 6 is present in both sets

View Answer

Answer: a

Explanation: – operator gives the set of elements in set a but not in set b.

21. What is the output of the following piece of code?

```
>>> a={5,6,7,8}
>>> b={7,8,10,11}
>>>a^b
```

a) {5,6,7,8,10,11}

b) {7,8}

c) Error as unsupported operand type of set data type

d) {5,6,10,11}

View Answer

Answer: d

Explanation: ^ operator returns a set of elements in set A or set B, but not in both (symmetric difference).

22. What is the output of the following code?

```
>>> s={5,6}
>>> s*3
```

a) Error as unsupported operand type for set data type

b) {5,6,5,6,5,6}

c) {5,6}

d) Error as multiplication creates duplicate elements which isn't allowed

View Answer

Answer: a

Explanation: The multiplication operator isn't valid for the set data type.

23. What is the output of the following piece of code?

```
>>> a={5,6,7,8}
>>> b={7,5,6,8}
>>> a==b
```

a) True

b) False

View Answer

Answer: a

Explanation: It is possible to compare two sets and the order of elements in both the sets doesn't matter if the values of the elements are the same.

24. What is the output of the following piece of code?

```
>>> a={3,4,5}
>>> b={5,6,7}
>>>a|b
```

a) Invalid operation

b) {3, 4, 5, 6, 7}

c) {5}

d) {3,4,6,7}

View Answer

Answer: d

Explanation: The operation in the above piece of code is union operation. This operation produces a set of elements in both set a and set b.

25. Is the following piece of code valid?

```
a={3,4,{7,5}}
print(a[2][0])
```

a) Yes, 7 is printed

b) Error, elements of a set can't be printed

c) Error, subsets aren't allowed

d) Yes, {7,5} is printed

View Answer

Answer: c

Explanation: In python, elements of a set must not be mutable and sets are mutable. Thus, subsets can't exist.

26. What is the output of the code shown?

```
s=set()
type(s)
```

a) <'set'>

b) <class 'set'>

c) set

d) class set

View Answer

Answer: b

Explanation: When we find the type of a set, the output returned is: .

27. The line of code shown below results in an error. State whether this statement is true or false.

```
s={2,3,4,[5,6]}
```

a) True

b) False

View Answer

Answer: a

Explanation: The set data type makes use of a principle known as hashing. This means that each item in the set should be hashable. Hashable in this context means immutable. List is mutable and hence the line of code shown above will result in an error.

28. Set makes use of _____

Dictionary makes use of _____

a) keys, keys

b) key values, keys

c) keys, key values

d) key values, key values

View Answer

Answer: c

Explanation: Set makes use of keys.

Dictionary makes use of key values.

29. Which of the following lines of code will result in an error?

a) s={abs}

b) s={4, 'abc', (1,2)}

c) s={2, 2.2, 3, 'xyz'}

d) s={san}

View Answer

Answer: d

Explanation: The line: s={san} will result in an error because 'san' is not defined. The line s={abs} does not result in an error because abs is a built-in function. The other sets shown do not result in an error because all the items are hashable.

30. What is the output of the code shown below?

```
s={2,5,6,6,7}
s
```

a) {2, 5, 7}

b) {2, 5, 6, 7}

c) {2, 5, 6, 6, 7}

d) Error

View Answer

Answer: b

Explanation: Duplicate values are not allowed in sets. Hence, the output of the code shown above will be a set containing the duplicate value only once. Therefore the output is: {2, 5, 6, 7}

EXPERIMENT -20

OBJECTIVE:

Write a Python class to reverse a string word by word.

SOURCECODE:

```
def reverseWords(input):

    # split words of string separated by space
    inputWords = input.split(" ")

    # reverse list of words
    # suppose we have list of elements list = [1,2,3,4],
    # list[0]=1, list[1]=2 and index -1 represents
    # the last element list[-1]=4 ( equivalent to list[3]=4 )
    # So, inputWords[-1::-1] here we have three arguments
    # first is -1 that means start from last element
    # second argument is empty that means move to end of list
    # third arguments is difference of steps
    inputWords=inputWords[-1::-1]

    # now join words with space
    output = ' '.join(inputWords)

    return output

if __name__ == "__main__":
    input = 'geeks quiz practice code'
    print reverseWords(input)
```

INPUT ANDOUTPUT:

"code practice quiz geeks"

VIVA Questions

1. Which of the following statements create a dictionary?

a) d = {}

b) d = {“john”:40, “peter”:45}

c) d = {40:”john”, 45:”peter”}

d) All of the mentioned

View Answer
Answer: d
Explanation: Dictionaries are created by specifying keys and values.

2. Read the code shown below carefully and pick out the keys?

```
1. d ={"john":40, "peter":45}
```

a) “john”, 40, 45, and “peter”

b) “john” and “peter”

c) 40 and 45

d) d = (40:”john”, 45:”peter”)

View Answer
Answer: b
Explanation: Dictionaries appear in the form of keys and values.

3. What will be the output?

```
1. d ={"john":40, "peter":45}
2. "john" in d
```

a) True

b) False

c) None

d) Error

View Answer
Answer: a
Explanation: In can be used to check if the key is int dictionary.

4. What will be the output?

```
1. d1 ={"john":40, "peter":45}
2. d2 ={"john":466, "peter":45}
3. d1 == d2
```

a) True

b) False

c) None

d) Error

View Answer
Answer: b
Explanation: If d2 was initialized as d2 = d1 the answer would be true.

5. What will be the output?

```
1. d1 ={"john":40, "peter":45}
2. d2 ={"john":466, "peter":45}
3. d1 > d2
```

a) True

b) False

c) Error

d) None

View Answer
Answer: c
Explanation: Arithmetic > operator cannot be used with dictionaries.
advertisement

6. What is the output?

```
1. d ={"john":40, "peter":45}
2. d["john"]
```

a) 40

b) 45

c) “john”

d) “peter”

View Answer
Answer: a
Explanation: Execute in the shell to verify.

7. Suppose d = {“john”:40, “peter”:45}, to delete the entry for “john” what command do we use

a) d.delete(“john”:40)

b) d.delete(“john”)

c) del d[“john”].

d) del d(“john”:40)

View Answer
Answer: c
Explanation: Execute in the shell to verify.

8. Suppose d = {“john”:40, “peter”:45}. To obtain the number of entries in dictionary which command do we use?

a) d.size()

b) len(d)

c) size(d)

d) d.len()

View Answer
Answer: b
Explanation: Execute in the shell to verify.

9. What will be the output?

```
1. d ={"john":40, "peter":45}
2. print(list(d.keys()))
```

a) [“john”, “peter”].

b) [“john”:40, “peter”:45].

c) (“john”, “peter”)

d) (“john”:40, “peter”:45)

View Answer
Answer: a
Explanation: The output of the code shown above is a list containing only keys of the dictionary d, in the form of a list.

10. Suppose d = {“john”:40, “peter”:45}, what happens when we try to retrieve a value using the expression d[“susan”]?

a) Since “susan” is not a value in the set, Python raises a KeyError exception

b) It is executed fine and no exception is raised, and it returns None

c) Since “susan” is not a key in the set, Python raises a KeyError exception

d) Since “susan” is not a key in the set, Python raises a syntax error

View Answer
Answer: c
Explanation: Execute in the shell to verify.

11. Which of these about a dictionary is false?

a) The values of a dictionary can be accessed using keys

b) The keys of a dictionary can be accessed using values

c) Dictionaries aren’t ordered

d) Dictionaries are mutable

View Answer
Answer: b
Explanation: The values of a dictionary can be accessed using keys but the keys of a dictionary can’t be accessed using values.

12. Which of the following is not a declaration of the dictionary?

a) {1: ‘A’, 2: ‘B’}

b) dict([[1,”A”],[2,”B”]])

c) {1,”A”,2”B”}

d) { }

View Answer
Answer: c
Explanation: Option c is a set, not a dictionary.

13. What is the output of the following code?

```
a={1:"A",2:"B",3:"C"}
for i,j in a.items():
print(i,j,end=" ")
```

a) 1 A 2 B 3 C

b) 1 2 3

c) A B C

d) 1:”A” 2:”B” 3:”C”

View Answer
Answer: a
Explanation: In the above code, variables i and j iterate over the keys and values of the dictionary respectively.

14. What is the output of the following piece of code?

```
a={1:"A",2:"B",3:"C"}
print(a.get(1,4))
```

a) 1

b) A

c) 4

d) Invalid syntax for get method

View Answer

Answer: b
Explanation: The get() method returns the value of the key if the key is present in the dictionary and the default value(second parameter) if the key isn't present in the dictionary.

15. What is the output of the following code?

```
a={1:"A",2:"B",3:"C"}
print(a.get(5,4))
```

a) Error, invalid syntax

b) A

c) 5

d) 4

View Answer
Answer: d
Explanation: The get() method returns the default value(second parameter) if the key isn't present in the dictionary.

16. What is the output of the following code?

```
a={1:"A",2:"B",3:"C"}
print(a.setdefault(3))
```

a) {1: 'A', 2: 'B', 3: 'C'}

b) C

c) {1: 3, 2: 3, 3: 3}

d) No method called setdefault() exists for dictionary

View Answer
Answer: b
Explanation: setdefault() is similar to get() but will set dict[key]=default if key is not already in the dictionary.

17. What is the output of the following code?

```
a={1:"A",2:"B",3:"C"}
a.setdefault(4,"D")
print(a)
```

a) {1: 'A', 2: 'B', 3: 'C', 4: 'D'}.

b) None.

c) Error.

d) [1,3,6,10].

View Answer
Answer: a
Explanation: setdefault() will set dict[key]=default if key is not already in the dictionary.

18. What is the output of the following code?

```
a={1:"A",2:"B",3:"C"}
b={4:"D",5:"E"}
a.update(b)
print(a)
```

a) {1: 'A', 2: 'B', 3: 'C'}

b) Method update() doesn't exist for dictionaries

c) {1: 'A', 2: 'B', 3: 'C', 4: 'D', 5: 'E'}

d) {4: 'D', 5: 'E'}

View Answer
Answer: c
Explanation: update() method adds dictionary b's key-value pairs to dictionary a. Execute in python shell to verify.

19. What is the output of the following code?

```
a={1:"A",2:"B",3:"C"}
b=a.copy()
b[2]="D"
print(a)
```

a) Error, copy() method doesn't exist for dictionaries

b) {1: 'A', 2: 'B', 3: 'C'}

c) {1: 'A', 2: 'D', 3: 'C'}

d) "None" is printed

View Answer
Answer: b
Explanation: Changes made in the copy of the dictionary isn't reflected in the original one.

20. What is the output of the following code?

```
a={1:"A",2:"B",3:"C"}
a.clear()
print(a)
```

a) None

b) { None:None, None:None, None:None}

c) {1:None, 2:None, 3:None}

d) { }

View Answer
Answer: d
Explanation: The clear() method clears all the key-value pairs in the dictionary.

21. Which of the following isn't true about dictionary keys?

a) More than one key isn't allowed

b) Keys must be immutable

c) Keys must be integers

d) When duplicate keys encountered, the last assignment wins

View Answer
Answer: c
Explanation: Keys of a dictionary may be any data type that is immutable.

22. What is the output of the following code?

```
a={1:5,2:3,3:4}
a.pop(3)
print(a)
```

a) {1: 5}

b) {1: 5, 2: 3}

c) Error, syntax error for pop() method

d) {1: 5, 3: 4}

View Answer
Answer: b
Explanation: pop() method removes the key-value pair for the key mentioned in the pop() method.

23. What is the output of the following code?

```
a={1:5,2:3,3:4}
print(a.pop(4,9))
```

a) 9

b) 3

c) Too many arguments for pop() method

d) 4

View Answer
Answer: a
Explanation: pop() method returns the value when the key is passed as an argument and otherwise returns the default value(second argument) if the key isn't present in the dictionary.

24. What is the output of the following code?

```
a={1:"A",2:"B",3:"C"}
for i in a:
print(i,end=" ")
```

a) 1 2 3

b) 'A' 'B' 'C'

c) 1 'A' 2 'B' 3 'C'

d) Error, it should be: for i in a.items():

View Answer
Answer: a
Explanation: The variable i iterates over the keys of the dictionary and hence the keys are printed.

25. Execute the following in Python shell?

```
>>> a={1:"A",2:"B",3:"C"}
>>>a.items()
```

a) Syntax error

b) dict_items([('A'), ('B'), ('C')])

c) dict_items([(1,2,3)])

d) dict_items([(1, 'A'), (2, 'B'), (3, 'C')])

View Answer
Answer: d
Explanation: The method items() returns list of tuples with each tuple having a key-value pair.

26. To open a file c:\scores.txt for reading, we use

a) infile = open("c:\scores.txt", "r")

b) infile = open("c:\\scores.txt", "r")

c) infile = open(file = "c:\scores.txt", "r")

d) infile = open(file = "c:\\scores.txt", "r")

View Answer
Answer: b
Explanation: Execute help(open) to get more details.

27. To open a file c:\scores.txt for writing, we use

a) outfile = open("c:\scores.txt", "w")

b) outfile = open("c:\\scores.txt", "w")

c) outfile = open(file = "c:\scores.txt", "w")

d) outfile = open(file = “c:\\scores.txt”, “w”)

View Answer
Answer: b
Explanation: w is used to indicate that file is to be written to.

28. To open a file c:\scores.txt for appending data, we use

a) outfile = open(“c:\\scores.txt”, “a”)

b) outfile = open(“c:\\scores.txt”, “rw”)

c) outfile = open(file = “c:\scores.txt”, “w”)

d) outfile = open(file = “c:\\scores.txt”, “w”)

View Answer
Answer: a
Explanation: a is used to indicate that data is to be appended.

29. Which of the following statements are true?

a) When you open a file for reading, if the file does not exist, an error occurs

b) When you open a file for writing, if the file does not exist, a new file is created

c) When you open a file for writing, if the file exists, the existing file is overwritten with the new file

d) All of the mentioned

View Answer
Answer: d
Explanation: The program will throw an error.

30. To read two characters from a file object infile, we use

a) infile.read(2)

b) infile.read()

c) infile.readline()

d) infile.readlines()

View Answer
Answer: a
Explanation: Execute in the shell to verify.

ADDITIONAL EXPERIMENTS

1. Python program to print "Hello Python"
2. Python program to do arithmetical operations
3. Python program to find the area of a triangle
4. Python program to solve quadratic equation

5. Python program to swap two variables
6. Python program to generate a random number
7. Python program to convert kilometers to miles
8. Python program to convert Celsius to Fahrenheit
9. Python program to display calendar
10. Python Program to Check if a Number is Positive, Negative or Zero
11. Python Program to Check if a Number is Odd or Even
12. Python Program to Check Leap Year
13. Python Program to Check Prime Number
14. Python Program to Print all Prime Numbers in an Interval
15. Python Program to Find the Factorial of a Number
16. Python Program to Display the multiplication Table
17. Python Program to Print the Fibonacci sequence
18. Python Program to Check Armstrong Number
19. Python Program to Find Armstrong Number in an Interval
20. Python Program to Find the Sum of Natural Numbers
21. Python Program to Find LCM
22. Python Program to Find HCF
23. Python Program to Convert Decimal to Binary, Octal and Hexadecimal
24. Python Program To Find ASCII value of a character
25. Python Program to Make a Simple Calculator
26. Python Program to Display Calendar
27. Python Program to Display Fibonacci Sequence Using Recursion
28. Python Program to Find Factorial of Number Using Recursion
29. Python Program to Add Two Matrices
30. Python Program to Multiply Two Matrices
31. Python Program to Transpose a Matrix
32. Python Program to Sort Words in Alphabetic Order
33. Python Program to Remove Punctuation From a String
34. Python program to copy all elements of one array into another array
35. Python program to find the frequency of each element in the array
36. Python program to left rotate the elements of an array
37. Python program to print the duplicate elements of an array

38. Python program to print the elements of an array
39. Python program to print the elements of an array in reverse order
40. Python program to print the elements of an array present on even position
41. Python program to print the elements of an array present on odd position
42. Python program to print the largest element in an array
43. Python program to print the smallest element in an array
44. Python program to print the number of elements present in an array
45. Python program to print the sum of all elements in an array
46. Python program to right rotate the elements of an array
47. Python program to sort the elements of an array in ascending order
48. Python program to sort the elements of an array in descending order
49. Python program to check if the given number is a Disarium Number
50. Python program to print all disarium numbers between 1 to 100
51. Python program to check if the given number is Happy Number
52. Python program to print all happy numbers between 1 and 100
53. Python program to determine whether the given number is a Harshad Number
54. Python program to print all pronic numbers between 1 and 100
55. Python program to create a Circular Linked List of N nodes and count the number of nodes
56. Python program to create a Circular Linked List of n nodes and display it in reverse order
57. Python program to create and display a Circular Linked List
58. Python program to delete a node from the beginning of the Circular Linked List
59. Python program to delete a node from the end of the Circular Linked List
60. Python program to delete a node from the middle of the Circular Linked List
61. Python program to find the maximum and minimum value node from a circular linked list
62. Python program to insert a new node at the beginning of the Circular Linked List
63. Python program to insert a new node at the end of the Circular Linked List
64. Python program to insert a new node at the middle of the Circular Linked List
65. Python program to remove duplicate elements from a Circular Linked List
66. Python program to search an element in a Circular Linked List
67. Python program to sort the elements of the Circular Linked List
68. Python program to convert a given binary tree to doubly linked list
69. Python program to create a doubly linked list from a ternary tree
70. Python program to create a doubly linked list of n nodes and count the number of nodes

71. Python program to create a doubly linked list of n nodes and display it in reverse order
72. Python program to create and display a doubly linked list
73. Python program to delete a new node from the beginning of the doubly linked list
74. Python program to delete a new node from the end of the doubly linked list
75. Python program to delete a new node from the middle of the doubly linked list
76. Python program to find the maximum and minimum value node from a doubly linked list
77. Python program to insert a new node at the beginning of the Doubly Linked list
78. Python program to insert a new node at the end of the Doubly Linked List
79. Python program to insert a new node at the middle of the Doubly Linked List
80. Python program to remove duplicate elements from a Doubly Linked List
81. Python program to rotate doubly linked list by N nodes

https://www.javatpoint.com/python-programs