

# Fundamentals of Python Programming 4<sup>th</sup> Semester CSE

## SYLLABUS

**UNIT:1 (08 Hours)**

**Introduction:** Installation, First Python Program: Interactive Mode Programming, Script Mode Programming; Identifiers, Reserved Words, Lines and Indentation, Multi-Line Statements, Quotation & Comments; Assigning Values to Variables, Multiple Assignment.

**Standard Data Types:** Numbers, Strings, Lists, Tuples, Dictionary; Data Type Conversion; Basic Operators: Arithmetic, Comparison, Assignment, Bitwise; Operators: Logical, Membership, Identity; Operators Precedence; Python Numbers & Mathematical functions, Python Strings.

### Multi Choice Question of UNIT I:

- Which of these is not a core data type?
  - Lists
  - Dictionary
  - Tuples
  - Class
- What will be the output of the following Python code?

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

  - he
  - lo
  - olleh
  - hello
- What data type is the object below?

```
L = [1, 23, 'hello', 1]
```

  - list
  - dictionary
  - array
  - tuple
- Which of the following results in a SyntaxError?
  - ""Once upon a time...", she said."
  - "He said, 'Yes!'"
  - '3\'
  - ""That's okay""
- What is the average value of the following Python code snippet?

```
>>>grade1 = 80
>>>grade2 = 90
>>>average = (grade1 + grade2) / 2
```

  - 85.0
  - 85.1
  - 95.0
  - 95.1
- What is the type of infinite?
  - Boolean
  - Integer
  - Float
  - Complex
- Which of the following is incorrect?
  - x = 0b101
  - x = 0x4f5
  - x = 19023
  - x = 03964
- What does  $3^4$  evaluate to?
  - 81
  - 12
  - 0.75
  - 7
- What will be the value of the following Python expression?

```
4 + 3 % 5
```

  - 4
  - 7
  - 2
  - 0
- What is the value of the following expression?

```
8/4/2, 8/(4/2)
```

- a) (1.0, 4.0)                      b) (1.0, 1.0)  
 c) (4.0, 1.0)                      d) (4.0, 4.0)
- 11.** What is the value of the following expression?  
 float(22//3+3/3)  
 a) 8                                      b) 8.0  
 c) 8.3                                  d) 8.33
- 12.** Is Python case sensitive when dealing with identifiers?  
 a) yes                                      b) no  
 c) machine dependent              d) none of the mentioned
- 13.** What is the maximum possible length of an identifier?  
 a) 31 characters                      b) 63 characters  
 c) 79 characters                      d) none of the mentioned
- 14.** Which of the following is an invalid variable?  
 a) my\_string\_1                      b) 1st\_string  
 c) foo                                      d) \_
- 15.** Which of the following is not a keyword?  
 a) eval                                      b) assert  
 c) nonlocal                              d) pass
- 16.** 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
- 17.** What will be the output of the following Python expression if x=15 and y=12?  
 x & y  
 a) b1101                                  b) 0b1101  
 c) 12                                      d) 1101
- 18.** What will be the output of the following Python expression?  
 0x35 | 0x75  
 a) 115                                      b) 116  
 c) 117                                      d) 118
- 19.** What will be the value of the following Python expression?  
 bin(10-2)+bin(12^4)  
 a) 0b10000                                  b) 0b10001000  
 c) 0b1000b1000                      d) 0b10000b1000
- 20.** What will be the output of the ~100 Python expression?  
 a) 101                                      b) -101  
 c) 100                                      d) -100

### Short Type Questions of UNITS I:

1. What is the difference between list and tuples in Python?
2. What are the key features of Python?
3. What type of language is python? Programming or scripting?
4. How is Python an interpreted language?
5. What is pep 8?
6. How is memory managed in Python?
7. What are python modules? Name some commonly used built-in modules in Python?
8. What are local variables and global variables in Python?
9. What is type conversion in Python?
10. Is indentation required in python?
11. What is the difference between Python Arrays and lists?
12. What are the built-in types of python?
13. Explain the need for Unicodes.

**Long Type Questions of Unit I:**

1. Explain the different string formats available in Python with examples.
2. Discuss the int(), float(), str(), chr() and complex() type conversion functions with examples.
3. Discuss the ord(), hex(), oct(), complex() and float() type conversion functions with examples.
4. Describe the **is** and **is not** operators and type() function. Also, discuss why Python is called as dynamic and strongly typed language.
5. What is Python? Describe its features and applications?
6. Differentiate between lists and tuples in Python?
7. Explain in detail about Python type conversion and type casting?
8. What are operators in Python? Describe specifically about identity membership operator?

**UNIT:2 (10 Hours)**

**Python Program Flow Control :** Conditional blocks using if, else and elif, Simple for loops in python, For loop using ranges, string, list and dictionaries, Use of while loops in python, Loop manipulation using pass, continue, break and else, Programming using Python conditional and loops block

**Python Functions, Modules And Packages:** Organizing python codes using functions, Organizing python projects into modules, Importing own module as well as external modules, Understanding Packages, Powerful Lamda function in python, Programming using functions, modules and external packages

**Python String, List And Dictionary Manipulations :** Building blocks of python programs, Understanding string in build methods, List manipulation using in build methods, Dictionary manipulation, Programming using string, list and dictionary in build functions

**Multi Choice Question of UNIT II:**

1. What will be the output of the following Python code snippet?  

```
k = [print(i) for i in my_string if i not in "aeiou"]
```

  - a) prints all the vowels in my\_string
  - b) prints all the consonants in my\_string
  - c) prints all characters of my\_string that aren't vowels
  - d) prints only on executing print(k)
2. What will be the output of the following Python code snippet?  

```
x = [i**+1 for i in range(3)]; print(x);
```

  - a) [0, 1, 2]
  - b) [1, 2, 5]
  - c) error, \*\*+ is not a valid operator
  - d) error, ',' is not allowed
3. What will be the output of the following Python code snippet?  

```
print([i+j for i in "abc" for j in "def"])
```

  - a) ['da', 'ea', 'fa', 'db', 'eb', 'fb', 'dc', 'ec', 'fc']
  - b) [['ad', 'bd', 'cd'], ['ae', 'be', 'ce'], ['af', 'bf', 'cf']]
  - c) [['da', 'db', 'dc'], ['ea', 'eb', 'ec'], ['fa', 'fb', 'fc']]
  - d) ['ad', 'ae', 'af', 'bd', 'be', 'bf', 'cd', 'ce', 'cf']
4. What will be the output of the following Python code?  

```
x = ['ab', 'cd']  
for i in x:  
    i.upper()  
print(x)
```

  - a) ['ab', 'cd']
  - b) ['AB', 'CD']
  - c) [None, None]
  - d) none of the mentioned
5. What will be the output of the following Python code?  

```
i = 1  
while True:  
    if i%2 == 0:
```

*break*

*print(i)*

*i += 2*

a) 1

b) 1 2

c) 1 2 3 4 5 6 ...

d) 1 3 5 7 9 11 ...

6. What will be the output of the following Python code?

*True = False*

*while True:*

*print(True)*

*break*

a) True

b) False

c) None

d) none of the mentioned

7. What will be the output of the following Python code?

*x = "abcdef"*

*while i in x:*

*print(i, end=" ")*

a) a b c d e f

b) abcdef

c) i i i i i ...

d) error

8. What will be the output of the following Python code?

*x = "abcdef"*

*i = "a"*

*while i in x:*

*x = x[:-1]*

*print(i, end = " ")*

a) i i i i i I

b) a a a a a a

c) a a a a a

d) none of the mentioned

9. What will be the output of the following Python code?

*for i in range(2.0):*

*print(i)*

a) 0.0 1.0

b) 0 1

c) error

d) none of the mentioned

10. What will be the output of the following Python code snippet?

*for i in [1, 2, 3, 4][::-1]:*

*print (i)*

a) 1 2 3 4

b) 4 3 2 1

c) error

d) none of the mentioned

11. What will be the output of the following Python code snippet?

*for i in ".join(reversed(list('abcd'))):*

*print (i)*

a) a b c d

b) d c b a

c) error

d) none of the mentioned

12. What will be the output of the following Python code?

*for i in range(10):*

*if i == 5:*

*break*

*else:*

*print(i)*

*else:*

*print("Here")*

a) 0 1 2 3 4 Here

b) 0 1 2 3 4 5 Here

c) 0 1 2 3 4

d) 1 2 3 4 5

13. What will be the output of the following Python code?

```
x = (i for i in range(3))
```

```
for i in x:
```

```
    print(i)
```

```
for i in x:
```

```
    print(i)
```

a) 0 1 2

b) error

c) 0 1 2 0 1 2

d) none of the mentioned

14. What will be the output of the following Python code snippet?

```
a = [0, 1, 2, 3]
```

```
for a[-1] in a:
```

```
    print(a[-1])
```

a) 0 1 2 3

b) 0 1 2 2

c) 3 3 3 3

d) error

15. What will be the output of the following Python statement?

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

a) a

b) ab

c) cd

d) dc

16. print(0xA + 0xB + 0xC):

a) 0xA0xB0xC

b) Error

c) 0x22

d) 33

17. What will be the output of the following Python code?

```
>>>max("what are you")
```

a) error

b) u

c) t

d) y

18. Suppose x is 345.3546, what is `format(x, "10.3f")` (\_ indicates space).

a) \_\_345.355 (2\_s)

b) \_\_345.355 (3\_s)

c) \_\_\_\_345.355 (4\_s)

d) \_\_\_\_345.354 (5\_s)

19. What will be the output of the following Python code?

```
print('*', "abcdef".center(7), '*')
```

a) \* abcdef \*

b) \* abcdef \*

c) \*abcdef \*

d) \* abcdef\*

20. What will be the output of the following Python code?

```
print("abcdef".center(9, '1'))
```

a) 11abcdef1

b) 1abcdef11

c) 1abcdef1

d) error

21. What will be the output of the following Python code?

```
print("xyyzxyzxyzxy".count('yy', 2))
```

a) 2

b) 0

c) 1

d) none of the mentioned

22. What will be the output of the following Python code?

```
elements = [0, 1, 2]
```

```
def incr(x):
```

```
    return x+1
```

```
print(list(map(incr, elements)))
```

a) [1, 2, 3]

b) [0, 1, 2]

c) error

d) none of the mentioned

23. What will be the output of the following Python code?

```
x = ['ab', 'cd']
```

```
print(list(map(len, x)))
```

- a) ['ab', 'cd']                      b) [2, 2]  
 c) ['2', '2']                        d) none of the mentioned

24. In \_\_\_\_\_ copy, the base address of the objects are copied. In \_\_\_\_\_ copy, the base address of the objects are not copied.

- a) deep. Shallow                      b) memberwise, shallow  
 c) shallow, deep                      d) deep, memberwise

25. What will be the output of the following Python code?

```
a = [1, 2, 3, 4, 5]
b = lambda x: (b(x[1:]) + x[:1]) if x else []
print(b(a))
```

- a) 1 2 3 4 5                              b) [5,4,3,2,1]  
 c) []                                        d) Error, lambda functions can't be called recursively

### Short Type Questions of UNITS II:

1. What are functions in Python?
2. What is `__init__`?
3. What is a lambda function?
4. What is self in Python?
5. How does break, continue and pass work?
6. What does `[::-1]` do?
7. How can you randomize the items of a list in place in Python?
8. What are python iterators?
9. How do you write comments in python?
10. What is pickling and unpickling?
11. What are docstrings in Python?
12. What is the purpose of is, not and in operators?
13. What is the usage of `help()` and `dir()` function in Python?
14. What is a dictionary in Python?
15. How can the ternary operators be used in python?
16. What does this mean: `*args`, `**kwargs`? And why would we use it?
17. What does `len()` do?
18. What are negative indexes and why are they used?
19. How to import modules in python?
20. What are Python libraries? Name a few of them.
21. What is module and package in Python?
22. How can you share global variables across modules?

### Long Type Questions of Unit II:

1. Illustrate the different types of control flow statements available in Python with flowcharts.
2. Write a Program to Prompt for a Score between 0.0 and 1.0. If the Score is out of range, print an error. If the score is between 0.0 and 1.0, print a grade using the following table

Score	Grade
$\geq 0.9$	A
$\geq 0.8$	B
$\geq 0.7$	C
$\geq 0.6$	D
$< 0.6$	F

3. Write a program to display the fibonacci sequences up to nth term where n is provided by the user.
4. Write a program to repeatedly check for the largest number until the user enters "done".
5. Write a program to find the sum of all Odd and Even numbers up to a number specified by the user.

6. Explain the need for **continue** and **break** statements. Write a program to check whether a number is prime or not. Prompt the user for input.
7. Describe the syntax for the following functions and explain with an example.  
a) abs() b) max() c) divmod() d) pow() e) len()
8. Write Pythonic code to solve the quadratic equation  $ax^2 + bx + c = 0$  by getting input for coefficients from the user.
9. Find the area and perimeter of a circle using functions. Prompt the user for input.
10. Write a Python program using functions to find the value of nPr and nCr without using inbuilt factorial() function.
11. Write a program to print the sum of the following series  $1 + 1/2 + 1/3 + \dots + 1/n$
12. Write a function which receives a variable number of strings as arguments. Find unique characters in each string.
13. Check if the items in the list are sorted in ascending or descending order and print suitable messages accordingly. Otherwise, print "Items in list are not sorted"
14. Write Pythonic code to multiply two matrices using nested loops and also perform transpose of the resultant matrix.
15. Write Python program to sort words in a sentence in decreasing order of their length. Display the sorted words along with their length
16. Write Pythonic code to create a function called most\_frequent that takes a string and prints the letters in decreasing order of frequency. Use dictionaries.

### UNIT:3 (12 Hours)

**Python File Operation :** Reading config files in python, Writing log files in python , Manipulating file pointer using seek, Programming using file operations, Understanding read functions, read(), readline() and readlines(), Understanding write functions, write() and writelines()

**Python Object Oriented Programming – Oops:** Concept of class, object and instances, Constructor, class attributes and destructors, Real time use of class in live projects, Inheritance , overlapping and overloading operators, Adding and retrieving dynamic attributes of classes , Programming using OOps support

**Python Regular Expression :** Powerful pattern matching and searching, Power of pattern searching using regex in python, Real time parsing of networking or system data using regex, Password, email, url validation using regular expression, Pattern finding programs using regular expression

**Python Exception Handling :** Avoiding code break using exception handling, Safe guarding file operation using exception handling, Handling and helping developer with error code, Programming using Exception handling

### Multi Choice Question of UNIT III:

1. To read the entire remaining contents of the file as a string from a file object infile, we use  

a) infile.read(2)	b) infile.read()
c) infile.readline()	d) infile.readlines()
2. Which are the two built-in functions to read a line of text from standard input, which by default comes from the keyboard?  

a) Raw_input & Input	b) Input & Scan
c) Scan & Scanner	d) Scanner
3. Which one of the following is not attributes of file?  

a) closed	b) softspace
c) rename	d) mode
4. What is the use of seek() method in files?  

a) sets the file's current position at the offset
b) sets the file's previous position at the offset

- c) sets the file's current position within the file  
d) none of the mentioned
5. What is the use of truncate() method in file?  
a) truncates the file size                      b) deletes the content of the file  
c) deletes the file size                      d) none of the mentioned
6. What is the pickling?  
a) It is used for object serialization                      b) It is used for object deserialization  
c) None of the mentioned                      d) All of the mentioned
7. What does the function re.match do?  
a) matches a pattern at the start of the string  
b) matches a pattern at any position in the string  
c) such a function does not exist  
d) none of the mentioned
8. What will be the output of the following Python code?  
`sentence = 'horses are fast'`  
`regex = re.compile('(P<animal>\w+) (P<verb>\w+) (P<adjective>\w+)')`  
`matched = re.search(regex, sentence)`  
`print(matched.group(2))`  
a) {'animal': 'horses', 'verb': 'are', 'adjective': 'fast'}  
b) ('horses', 'are', 'fast')  
c) 'horses are fast'  
d) 'are'
9. The character Dot (that is, '.') in the default mode, matches any character other than \_\_\_\_\_  
a) caret                      b) ampersand  
c) percentage symbol                      d) newline
10. The expression a{5} will match \_\_\_\_\_ characters with the previous regular expression.  
a) 5 or less                      b) exactly 5  
c) 5 or more                      d) exactly 4
11. \_\_\_\_\_ matches the start of the string. \_\_\_\_\_ matches the end of the string.  
a) '^', '\$'                      b) '\$', '^'  
c) '\$', '?'                      d) '?', '^'
12. What will be the output of the following Python function?  
`re.findall("hello world", "hello", 1)`  
a) ['hello']                      b) []  
c) hello                      d) hello world
13. What will be the output of the following Python code?  
`re.sub('morning', 'evening', 'good morning')`  
a) 'good evening'                      b) 'good'  
c) 'morning'                      d) 'evening'
14. What will be the output of the following Python code?  
`re.split('[a-c]', '0a3B6', re.I)`  
a) Error                      b) ['a', 'B']  
c) ['0', '3B6']                      d) ['a']
15. Which of the following pattern matching modifiers permits whitespace and comments inside the regular expression?  
a) re.L                      b) re.S  
c) re.U                      d) re.X
16. Which of the following special characters matches a pattern only at the end of the string?  
a) \B                      b) \X  
c) \Z                      d) \A



17. Which of the following special characters represents a comment (that is, the contents of the parenthesis are simply ignored)?
- (?:...)
  - (?=...)
  - (?!...)
  - (?#...)
18. \_\_\_\_\_ represents an entity in the real world with its identity and behaviour.
- A method
  - An object
  - A class
  - An operation
19. What is setattr() used for?
- To access the attribute of the object
  - To set an attribute
  - To check if an attribute exists or not
  - To delete an attribute
20. The assignment of more than one function to a particular operator is \_\_\_\_\_
- Operator over-assignment
  - Operator overriding
  - Operator overloading
  - Operator instance
21. What are the methods which begin and end with two underscore characters called?
- Special methods
  - In-built methods
  - User-defined methods
  - Additional methods
22. What is hasattr(obj,name) used for?
- To access the attribute of the object
  - To delete an attribute
  - To check if an attribute exists or not
  - To set an attribute
23. Which of these is not a fundamental feature of OOP?
- Encapsulation
  - Inheritance
  - Instantiation
  - Polymorphism
24. Methods of a class that provide access to private members of the class are called as \_\_\_\_\_ and \_\_\_\_\_
- getters/setters
  - \_\_repr\_\_/\_str\_\_
  - user-defined functions/in-built functions
  - \_\_init\_\_/\_del\_\_
25. How many except statements can a try-except block have?
- zero
  - one
  - more than one
  - more than zero
26. When will the else part of try-except-else be executed?
- always
  - when an exception occurs
  - when no exception occurs
  - when an exception occurs in to except block
27. Can one block of except statements handle multiple exception?
- yes, like except TypeError, SyntaxError [...]
  - yes, like except [TypeError, SyntaxError]
  - no
  - none of the mentioned
28. What happens when '1' == 1 is executed?
- we get a True
  - we get a False
  - an TypeError occurs
  - a ValueError occurs
29. What will be the output of the following Python code?
- ```
g = (i for i in range(5))
type(g)
```
- class <'loop'>
  - class <'iteration'>
  - class <'range'>
  - class <'generator'>
30. What will be the output of the following Python code?
- ```
lst = [1, 2, 3]
lst[3]
```
- NameError
  - ValueError
  - IndexError
  - TypeError

**Short Type Questions of UNITS III:**

1. Explain split(), sub(), subn() methods of “re” module in Python.
2. What are Python packages?
3. How can files be deleted in Python?
4. Does Python have OOps concepts?
5. Explain Inheritance in Python with an example.
6. How are classes created in Python?
7. Does python support multiple inheritance?
8. What is Polymorphism in Python?
9. Define encapsulation in Python?
10. How do you do data abstraction in Python?
11. Does python make use of access specifiers?
12. What does an object() do?

**Long Type Questions of Unit III:**

1. Write Python Program to Reverse Each Word in “secret\_societies.txt” file
2. Write Python Program to Count the Occurrences of Each Word and Also Count the Number of Words in a “quotes.txt” File.
3. Write Python Program to Find the Longest Word in a File. Get the File Name from User.
4. Discuss the following methods supported by compiled regular expression objects.  
a) search() b) match() c) findall()
5. Consider a line “From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008” in the file email.txt. Write Pythonic code to read the file and extract email address from the lines starting from the word “From”. Use regular expressions to match email address.
6. Describe the need for catching exceptions using try and except statements
7. Write Python program to calculate the Arc Length of
8. an Angle by assigning values to the radius and angle data attributes of the class ArcLength.
9. Write Python Program to simulate a Bank Account with support for depositMoney, withdrawMoney and showBalance Operations.
10. Given three Points (x1, y1), (x2, y2) and (x3, y3), write a Python program to check if they are Collinear.
11. Discuss inheritance in Python programming language. Write a Python program to demonstrate the use of super() function.
12. Program to demonstrate the Overriding of the Base Class method in the Derived Class.
13. Write Python program to demonstrate Multiple Inheritance.
14. Given the Coordinates (x, y) of a center of a Circle and its radius, write Python program to determine whether the Point lies inside the Circle, on the Circle or outside the Circle.
15. Write Python Program to Demonstrate Multiple Inheritance with Method Overriding.
16. Write Pythonic code to overload “+”, “-” and “\*” operators by providing the methods \_\_add\_\_, \_\_sub\_\_ and \_\_mul\_\_.
17. Write Pythonic code to create a function named **move\_rectangle()** that takes an object of **Rectangle** class and two numbers named **dx** and **dy**. It should change the location of the Rectangle by adding dx to the x coordinate of corner and adding dy to the y coordinate of corner.
18. What is Exception? Explain Exception as control flow mechanism with suitable example.

**UNIT:4 (15 Hours)**

**Python Database Interaction:** SQL Database connection using python, Creating and searching tables, Reading and storing config information on database, Programming using database connections

**Python Multithreading:** Understanding threads, forking threads, Synchronizing the threads, Programming using multithreading

**Python CGI Introduction :** Writing python program for CGI application, Creating menus and accessing files, Server client program

#### **Multi Choice Question of UNIT IV:**

1. What does a threading.Lock do?
  - a) Synchronize threads
  - b) Pass messages between threads
  - c) Wait until a thread is finished
  - d) Allow only one thread at a time to access a resource
2. How many CPUs (or cores) will the Python threading library take advantage of simultaneously?
  - a) Two
  - b) One
  - c) None
  - d) All of the available CPUs
3. What does the Thread.join() method do?
  - a) Merges two threads into one
  - b) Restricts access to a resource
  - c) Waits for the thread to finish
  - d) Adds the thread to a pool
4. Race conditions :
  - a) Which thread will access the shared resources
  - b) Two threads incorrectly accessing a shared resource
  - c) Testing which thread completes first
  - d) execution of critical section
5. How to detect the status of a python thread?
  - a) isAlive()
  - b) isActive()
  - c) None
  - d) isDaemon()
6. Which Python library runs a function as thread?
  - a) None
  - b) thread
  - c) threading
  - d) \_threading
7. What is the difference between a semaphore and bounded semaphore?
  - a) Semaphore holds a counter for the number of release() calls minus the number of acquire() calls, plus an initial value but bounded semaphore doesn't.
  - b) Bounded semaphore holds a counter for the number of release() calls minus the number of acquire() calls, plus an initial value but semaphore doesn't.
  - c) A bounded semaphore makes sure its current value doesn't exceed its initial value while semaphore doesn't.
  - d) A semaphore makes sure its current value doesn't exceed its initial value while bounded semaphore doesn't.
8. Which synchronization method is used to guard the resources with limited capacity, e.g. a database server?
  - a) Lock
  - b) Semaphore
  - c) Condition
  - d) Event
9. What is the exception raised for an error that doesn't fall in any of the categories?
  - a) LookupError
  - b) SystemError
  - c) RuntimeError
  - d) ReferenceError
10. What would be the impact of multithreading on a uni-processor system?
  - a) Increase throughput
  - b) Degrade performance
  - c) Reduce execution time
  - d) Improve performance
11. Which one is reentrant lock type?
  - a) Condition
  - b) Lock
  - c) Semaphore
  - d) RLock
12. Which method is used to identify a thread?
  - a) getName()
  - b) None
  - c) getThread()
  - d) get\_ident()
13. What is the method that wakes up all thread waiting for the condition?
  - a) notify()
  - b) releaseAll()
  - c) release()
  - d) notifyAll()

14. Config() in Python Tkinter are used for
  - a) destroy the widget
  - b) place the widget
  - c) change property of the widget
  - d) configure the widget
15. Correct way to draw a line in canvas tkinter ?
  - a) line()
  - b) canvas.create\_line()
  - c) create\_line(canvas)
  - d) None of the above
16. Creating line are come in which type of thing ?
  - a) GUI
  - b) Canvas
  - c) Both of the above
  - d) None of the above
17. Essential thing to create a window screen using tkinter python?
  - a) call tk() function
  - b) create a button
  - c) To define a geometry
  - d) All of the above
18. For user Entry data, which widget we use in tkinter ?
  - a) Entry
  - b) Text
  - c) Both of the above
  - d) None of the above
19. How pack() function works on tkinter widget ?
  - a) According to x,y coordinate
  - b) According to row and column vise
  - c) According to left,right,up,down
  - d) None of the above
20. How we import a tkinter in python program ?
  - a) import tkinter
  - b) import tkinter as t
  - c) from tkinter import \*
  - d) All of the above
21. In which of the following field, we can put our Button?
  - a) Window
  - b) Frame
  - c) Label
  - d) All of the above
22. Minimum number of argument we pass in a function to create a rectangle using canvas tkinter ?
  - a) 2
  - b) 4
  - c) 6
  - d) 5
23. To change the color of the text in the Button widget, what we use ?
  - a) bg
  - b) fg
  - c) color
  - d) cchng
24. To delete any widget from the screen which function we use ?
  - a) stop()
  - b) delete()
  - c) destroy()
  - d) break()
25. what is Tk() in tkinter python ?
  - a) It is function
  - b) It is constructor
  - c) It is widget
  - d) All of the above

#### **Short Type Questions of UNITS IV:**

1. What is a thread in Python?
2. What is multithreading?
3. How is Multithreading achieved in Python?
4. What's the difference between Python threading and multiprocessing?
5. How are Python multithreading and multiprocessing related?
6. Explain different methods for threads?
7. Explain the thread synchronization process?
8. What are sockets?
9. How do you create a socket object in Python?
10. How many types of socket methods available in Python's socket library?
11. What are the methods available for the server socket?
12. What are the methods available for the client socket?
13. What are the general socket methods available in Python?
14. Write a program to draw a Circle in Python using Turtle?

**15.** Write a program to draw a rectangle in Python using Turtle?

**Long Type Questions of Unit IV:**

1. Write a program to draw overlapping circles inclined to 30 degrees on the computer screen?
2. How to do logo design with Turtle?
3. Write a program to display multiplication table of 5 and 7 using multi-threading in Python?
4. Explain about Thread? Describe the advantage with an example?
5. What is Synchronization? Explain the Synchronization in Python with example?
6. Write a brief note on pen control methods?
7. Write a script to draw the following pattern using Turtle?

