

## MCQ

1. Who developed Python Programming Language?

- a) Wick van Rossum
- b) Rasmus Lerdorf
- c) Guido van Rossum**
- d) Niene Stom

2. Which type of Programming does Python support?

- a) object-oriented programming
- b) structured programming
- c) functional programming
- d) all of the mentioned**

3. Is Python case sensitive when dealing with identifiers?

- a) no
- b) yes**
- c) machine dependent
- d) none of the mentioned

4. Which of the following is the correct extension of the Python file?

- a) .python
- b) .pl
- c) .py**
- d) .p

5. Is Python code compiled or interpreted?

- a) Python code is both compiled and interpreted**
- b) Python code is neither compiled nor interpreted
- c) Python code is only compiled
- d) Python code is only interpreted

6. All keywords in Python are in \_\_\_\_\_

- a) Capitalized
- b) lower case
- c) UPPER CASE
- d) None of the mentioned**

7. What will be the value of the following Python expression?

**4** + **3** % **5**

- a) 7
- b) 2
- c) 4
- d) 1

8. Which of the following is used to define a block of code in Python language?

- a) **Indentation**
- b) Key
- c) Brackets
- d) All of the mentioned

9. Which keyword is used for function in Python language?

- a) Function
- b) **Def**
- c) Fun
- d) Define

10. Which of the following character is used to give single-line comments in Python?

- a) //
- b) **#**
- c) !
- d) /\*

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

```
i = 1
while True:
    if i%3 == 0:
        break
    print(i)

    i += 1
```

- a) 1 2 3
- b) **error**
- c) 1 2
- d) none of the mentioned

12. Which of the following functions can help us to find the version of python that we are currently working on?

- a) sys.version(1)
- b) sys.version(0)

- c) `sys.version()`
- d) **`sys.version`**

13. Python supports the creation of anonymous functions at runtime, using a construct called \_\_\_\_\_

- a) `pi`
- b) `anonymous`
- c) **`lambda`**
- d) none of the mentioned

14. What is the order of precedence in python?

- a) Exponential, Parentheses, Multiplication, Division, Addition, Subtraction
- b) Exponential, Parentheses, Division, Multiplication, Addition, Subtraction
- c) Parentheses, Exponential, Multiplication, Division, Subtraction, Addition
- d) **Parentheses, Exponential, Multiplication, Division, Addition, Subtraction**

15. What will be the output of the following Python code snippet if `x=1`?

```
x<<2
```

- a) **4**
- b) 2
- c) 1
- d) 8

16. What does `pip` stand for python?

- a) unlimited length
- b) all private members must have leading and trailing underscores
- c) **Preferred Installer Program**
- d) none of the mentioned

17. Which of the following is true for variable names in Python?

- a) underscore and ampersand are the only two special characters allowed
- b) **unlimited length**
- c) all private members must have leading and trailing underscores
- d) none of the mentioned

18. What are the values of the following Python expressions?

```
2**(3**2)
(2**3)**2
2*3**2
```

**a) 512, 64, 512**

b) 512, 512, 512

c) 64, 512, 64

d) 64, 64, 64

19. Which of the following is the truncation division operator in Python?

a) |

**b) //**

c) /

d) %

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

```
l=[1, 0, 2, 0, 'hello', ", []]  
list(filter(bool, l))
```

a) [1, 0, 2, 'hello', ", []]

b) Error

**c) [1, 2, 'hello']**

d) [1, 0, 2, 0, 'hello', ", []]

21. Which of the following functions is a built-in function in python?

a) factorial()

**b) print()**

c) seed()

d) sqrt()

22. Which of the following is the use of id() function in python?

a) Every object doesn't have a unique id

**b) Id returns the identity of the object**

c) All of the mentioned

d) None of the mentioned

23. The following python program can work with \_\_\_\_ parameters.

```
def f(x):  
    def f1(*args, **kwargs):  
        print("Sanfoundry")  
        return x(*args, **kwargs)  
    return f1
```

**a) any number of**

b) 0

c) 1

d) 2

23. The following python program can work with \_\_\_\_ parameters.

```
def f(x):  
    def f1(*args, **kwargs):  
        print("Sanfoundry")  
        return x(*args, **kwargs)  
    return f1
```

**a) any number of**

b) 0

c) 1

d) 2

24. What will be the output of the following Python function?

```
min(max(False,-3,-4), 2,7)
```

a) -4

b) -3

c) 2

**d) False**

25. Which of the following is not a core data type in Python programming?

a) Tuples

b) Lists

**c) Class**

d) Dictionary

26. What will be the output of the following Python expression if x=56.236?

```
print("%.2f"%x)
```

a) 56.236

b) 56.23

c) 56.0000

**d) 56.24**

27. Which of these is the definition for packages in Python?

a) A set of main modules

**b) A folder of python modules**

c) A number of files containing Python definitions and statements

d) A set of programs making use of Python modules

28. What will be the output of the following Python function?

```
len(["hello",2,4,6])
```

- a) Error
- b) 6
- c) 4**
- d) 3

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

```
x = 'abcd'
for i in x:
    print(i.upper())
```

- a) a B C D
- b) a b c d
- c) error
- d) A B C D**

30. What is the order of namespaces in which Python looks for an identifier?

- a) Python first searches the built-in namespace, then the global namespace and finally the local namespace
- b) Python first searches the built-in namespace, then the local namespace and finally the global namespace
- c) Python first searches the local namespace, then the global namespace and finally the built-in namespace**
- d) Python first searches the global namespace, then the local namespace and finally the built-in namespace

## **PROBLEM**

### **1. Python Program to Compute the Power of a Number**

```
import math
base=4
exponent=2
result = pow(base,exponent)
print(result)
```

```
In [3]: base=4
...: exponent=2
...: result = pow(base,exponent)
...: print(result)
16
```

## 2. Python Program to Find LCM

---

```
def calculate_lcm(x, y):
    if x > y:
        greater = x
    else:
        greater = y
    while(True):
        if((greater % x == 0) and (greater % y == 0)):
            lcm = greater
            break
        greater += 1
    return lcm
num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
print("The L.C.M. of", num1,"and", num2,"is", calculate_lcm(num1, num2))
```

```
...: num1 = int(input("Enter first number: "))
...: num2 = int(input("Enter second number: "))
...: print("The L.C.M. of", num1,"and", num2,"is",
calculate_lcm(num1, num2))
Enter first number: 4
Enter second number: 6
The L.C.M. of 4 and 6 is 12
```

## 3. Python Program to check if given array is Monotonic

---

```
def is_Monotonic(arr):
    return (all(arr[i] <= arr[i + 1] for i in range(len(arr) - 1)) or
            all(arr[i] >= arr[i + 1] for i in range(len(arr) - 1)))
arr = [10,9,6,4,2]
is_Monotonic(arr)
```

```

...:         all(arr[i] >= arr[i+1]
range(len(arr) - 1))
...: arr = [10,9,6,4,2]
...: is_Monotonic(arr)
Out[5]: True

```

#### 4. Python program to print all negative numbers in a range

```

def is_negative(a,b):
    for num in range(a, b + 1):
        if num < 0:
            print(num, end = " ")

a=int(input("Enter start of the range: "))
b=int(input("Enter end of the range: "))
is_negative(a,b)

```

```

...: a=int(input("Enter start of the range: "))
...: b=int(input("Enter end of the range: "))
...: is_negative(a,b)
Enter start of the range: -6
Enter end of the range: 2
-6 -5 -4 -3 -2 -1

```

#### 5. A. Python program to split and join a string

```

s = 'Life Science Batch'
print(s.split(" "))
print("&".join(s.split()))

```

```

...: print(s.split(" "))
...: print("&".join(s.split()))
['Life', 'Science', 'Batch']
Life&Science&Batch

```

#### B. Python | Check if a given string is binary string or not

```

def check(string):
    try:
        int(string, 2)
    except ValueError:
        return "No"
    return "Yes"

```



```
string1='101001010'  
print(check(string1))
```

```
...:         return "Yes"  
...:  
...: string1='101001010'  
...: print(check(string1))  
Yes
```

#### 6. A. Python program to convert time from 12 hour to 24 hour format

```
def convert(string):  
    if string[-2:] == "AM" and string[:2] == "12":  
        return "00" + string[2:-2]  
    elif string[-2:] == "AM":  
        return string[:-2]  
    elif string[-2:] == "PM" and string[:2] == "12":  
        return string[:-2]  
    else:  
        return str(int(string[:2]) + 12) + string[2:8]  
time="01:58:42PM"  
print("12-hour Format time:: ", time)  
print("24-hour Format time ::",convert(time))
```

```
...: print("12-hour Format time:: ", time)  
...: print("24-hour Format time ::",convert(time))  
12-hour Format time::  01:58:42PM  
24-hour Format time :: 13:58:42
```

#### B. Python program to find difference between current time and given time

```
import datetime  
today = datetime.datetime.now()  
print(today)  
given = datetime.datetime(2022,8,25,11,59)  
print(given)  
time_diff = given - today  
print("Time Difference:",time_diff)
```

```
...: time_diff = given - today
...: print("Time Difference:",time_diff)
2022-06-13 16:29:51.741907
2022-08-25 11:59:00
Time Difference: 72 days, 19:29:08.258093
```