Python-PrepTerm
Quiz

Code:	MT2020131

1. What is output for:

   a = ['hat', 'mat', 'rat']

   'rhyme'.join(a)

   1. `['hat','mat','rat','rhyme']`
   2. `'hatmatratrhyme'`
   3. `['hat mat rat rhyme']`
   4. `'hatrhymematrhyme rat'`

2. What is the following function sorts a list?

   1. **list** . reverse ()
   2. **list** . sort ([func])
   3. **list** . pop(obj=**list**[−1])
   4. **list** . remove(obj)

3. How to create a frame in Python?

   1. Frame = new.window()
   2. Frame = frame.new()
   3. Frame = Frame()
   4. Frame = window.new()

4. Which of the following function convert a string to a float in python?

   1. **int**(x [, base])
   2. **long**(x [,base])
   3. **float**(x)
   4. **str**(x)

5. Which of the following is required to create a new instance of the class?

   1. A constructor
   2. A class

3. A value-returning method
4. A None method

6. Name the error that doesn't cause program to stop/end, but the output is not the desired result or is incorrect.

   1. Syntax error
   2. Runtime error
   3. Logical error
   4. All of the above

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

```
class Sales:
    def __init__(self, id):
        self.id = id
        id = 100

val = Sales(123)
print (val.id)
```

   1. SyntaxError, this program will not run
   2. 100
   3. 123
   4. None of the above

8. What is the output of the following code?

```
eval("1 + 3 * 2")
```

   1. 1+6
   2. 4*2
   3. 1+3*2
   4. 7

9. What happens in the below code?

```
class A:
   def __init__(self, i=100):
       self.i=i
class B(A):
   def __init__(self,j=0):
       self.j=j
def main():
   b= B()
   print(b.i)
   print(b.j)
main()
```

   1. Class B inherits all the data fields of class A.

2. Class B needs an Argument.
3. The data field `'j'` cannot be accessed by object b.
4. Class B is inheriting class A but the data field `'i'` in A cannot be inherited.

10. Which of the following function converts a string to all lowercase?

    1. lower()
    2. lstrip ()
    3. **max**(**str**)
    4. **min**(**str**)

11. What is the following function removes an object from a list?

    1. **list** .index(obj)
    2. **list** . insert (index, obj)
    3. **list** .pop(obj=**list**[−1])
    4. **list** .remove(obj)

12. For tuples and list which is correct?

    1. List and tuples both are mutable.
    2. List is mutable whereas tuples are immutable.
    3. List and tuples both are immutable.
    4. List is immutable whereas tuples are mutable.

13. What is the output of **print** tinylist ∗ 2 **if** tinylist = [123, `'john'`]?

    1. [123, `'john'`, 123, `'john'`]\ lstinline
    2. [123, `'john'`] ∗ 2\ lstinline
    3. Error
    4. None of the above.

14. Which of the following function of dictionary gets all the keys from the dictionary?

    1. getkeys()
    2. key()
    3. keys()
    4. None of the above.

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

    1. llo World!
    2. H
    3. llo
    4. None of the above.

16. What is output of following code:

```
num=3
while True:
    if (num%0o12 == 0):
        break
print(num)
num += 1
```

    1. 3 4 5 6 7 8 9 10 11 12
    2. 3 4 5 6 7 8 9
    3. 3 4 5 6 7 8 9 10 11
    4. None of the above

17. There are different basic operators in python and work according to the order of their precedence.

    Arrange the order of precedence of the following operators:

    1. Division
    2. Multiplication
    3. Parentheses
    4. Exponential
    5. Addition
    6. Subtraction

    1. i, ii, iii, iv, v, vi.
    2. iv, iii, ii, i, vi, v.
    3. iii, iv, i, ii, v, vi.
    4. iv, iii, i, ii, v, vi.

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

    1. Hello World!Hello World!
    2. Hello World! $*$ 2
    3. Hello World!
    4. None of the above.

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

```
for i in ['t', 'n', 'i ', 'o', 'p'][::-1]:
    print(i)
```

    1. t n i o p
    2. p o i n t
    3. t n i o p 1 0 −1
    4. p o i n t 1 0 −1

20. When is the finally block executed?

    1. when there is no exception
    2. when there is an exception
    3. only if some condition that has been specified is satisfied
    4. always