

# Python-PrepTerm

## Quiz

<b>Code:</b>	MT2020054
--------------	-----------

1. What should be given in range of the given below code to print nothing in output?

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

1. 0.1
  2. 0
  3. NULL
  4. 1
2. Which of the following function convert a String to a list in python?

1. **repr**(x)
2. **eval**(str)
3. **tuple**(s)
4. **list**(s)

3. 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.

4. 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

5. Which of the following environment variable for Python contains the path of an initialization file containing Python source code?

1. PYTHONPATH
  2. PYTHONSTARTUP
  3. PYTHONCASEOK
  4. PYTHONHOME
6. 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)`
7. Which of the following operator in python evaluates to true if the variables on either side of the operator point to the same object and false otherwise?
1. `**`
  2. `//`
  3. `is`
  4. `not in`
8. Using the pack manager, how you can you put the components in a container in the same row?
1. `Component.pack(side= 'LEFT')`
  2. `Component.pack('Left')`
  3. `Component.pack(side=LEFT)`
  4. `Component.pack(Left=side)`
9. Which of the following statements are correct about the given code snippet?
- ```

class A:
    def _init_(self , i = 0):
        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 A, but the data field 'i' in A is not inherited.
  2. Class B inherits A, thus automatically inherits all data fields in A.
  3. When you create an object of B, you have to pass an argument such as B(5).

4. The data field 'j' cannot be accessed by object b.
10. 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.
11. `nfig()` in Python Tkinter are used for
1. destroy the widget
  2. place the widget
  3. change property of the widget
  4. configure the widget
12. 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
13. What will be the output of the following Python code?
- ```
def foo(): try: return 1 finally: return 2 k = foo() print(k)
```
1. 1
  2. 2
  3. 3
  4. error, there is more than one return statement in a single try-finally block
14. Analyze the code:
- ```
print("Recursive Function")
def factorial(n):
    return(n*factorial(n-1))
factorial(4)
```
1. Recursive Function 24.
  2. Recursive Function.
  3. Function runs infinitely and causes a StackOverflowError.

4. Syntax Error.
15. 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)`
16. What is the following function reverses objects of list in place?
1. `list.reverse()`
  2. `list.sort([func])`
  3. `list.pop(obj=list[-1])`
  4. `list.remove(obj)`
17. 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)`
18. What will be the output of the following Python code?
- ```
try:
    if '1' != 1:
        raise "someError"
    else:
        print("someError has not occurred")
except "someError":
    print ("someError has occurred")
```
1. someError has occurred
  2. someError has **not** occurred
  3. invalid code
  4. none of the mentioned
19. What is the following function returns item from the list with max value?
1. `cmp(list)`
  2. `len(list)`
  3. `max(list)`
  4. `min(list)`
20. Is the following Python code valid?

```
try :  
    # Do something  
except :  
    # Do something  
finally :  
    # Do something
```

1. no, there is no such thing as finally
2. no, finally cannot be used with except
3. no, finally must come before except
4. yes