Python-PrepTerm
Quiz

Code:	MT2020076

1. 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)
2. What is output of following code:

   a = (1, 2) a[0] +=1
    1. (1,1,2)
    2. 2
    3. Type Error
    4. Syntax Error
3. What is the following function gives the total length of the list?
    1. **cmp**(**list**)
    2. **len**(**list**)
    3. **max**(**list**)
    4. **min**(**list**)
4. 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
5. 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

6. 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**

7. 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)

8. 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.

9. Which of the following statements can be used to check, whether an object obj is an instance of class A or not?

    1. obj.**isinstance**(A)
    2. A.**isinstance**(obj)
    3. **isinstance**(obj, A)
    4. **isinstance**(A, obj)

10. 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

11. What is output for **min**("hello world")

    1. e
    2. a blank space character
    3. w
    4. None of the above.

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

```
minidict = { 'name': 'TutorialsPoint', 'name': 'website'}
print(minidict['name'])
```

    1. TutorialsPoint
    2. Website
    3. ('TutorialsPoint', 'website')
    4. It will show an Error.

13. 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

14. Which of the following operator in python evaluates to true if it does not finds a variable in the specified sequence and false otherwise?

    1. **
    2. //
    3. **is**
    4. **not in**

15. Syntax error in python is detected by _____ at _____

    1. compiler/ compile time
    2. interpreter/ run time
    3. compiler/ run time
    4. interpreter/ compile time

16. How many except statements can a try-except block have?

    1. zero
    2. one
    3. more than one
    4. more than zero

17. 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.

18. What is the following function compares elements of both dictionaries dict1, dict2?

    1. dict1.**cmp**(dict2)
    2. dict1.sort(dict2)
    3. **cmp**(dict1, dict2)
    4. None of the above.

19. 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)

20. 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.