Python-PrepTerm
Quiz

Code:	MT2020033

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. How many except statements can a try-except block have?
    1. zero
    2. one
    3. more than one
    4. more than zero
3. 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.
4. 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
5. 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

6. 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.

7. Which of the following function sets the integer starting value used in generating random numbers?

1. choice(seq)
2. randrange ([start ,] stop [, step])
3. random()
4. seed([x])

8. What will be the output of the below given code?

```
colors = ["white", "Black", "Grey"]
x = "Red" not in colors
```

1. Yes
2. No
3. Error: not in not defined
4. True

9. 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**

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

```
print(type(1/2))
```

1. <**class** 'float'>
2. <**class** 'int'>

3. NameError: `'1/2'` **is not** defined.
4. 0.5

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

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

12. 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.

13. 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

14. 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

15. 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

16. 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.

17. 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

18. 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**

19. 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.

20. 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.