Python-PrepTerm
Quiz

Code:	MT2020038

1. 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])

2. 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.

3. 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'`

4. 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

5. 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.

6. What is the output for:

```
'you are doing well'[2:999]
```

   1. 'you are doing well'
   2. ' '
   3. Index error.
   4. 'u are doing well'

7. 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.

8. 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)

9. What is output of following code:

    a = (1, 2) a[0] +=1

    1. (1,1,2)
    2. 2
    3. Type Error
    4. Syntax Error

10. Which of the following environment variable for Python is an alternative module search path?

    1. PYTHONPATH
    2. PYTHONSTARTUP
    3. PYTHONCASEOK
    4. PYTHONHOME

11. What is the following function gives the total length of the list?

    1. **cmp**(**list**)
    2. **len**(**list**)
    3. **max**(**list**)
    4. **min**(**list**)

12. 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.

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

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

15. rrect way to draw a line in canvas tkinter?

    1. line()
    2. canvas.create_line()
    3. create_line(canvas)
    4. None of the above

16. What is the output of the following code?

```
def nprint(message, n):
while(n > 0):
    print(message)
n-=1
nprint('z', 5)
```

    1. zzzz
    2. zzzzz
    3. Syntax Error
    4. Infinite Loop

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

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

18. 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

19. 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.

20. 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.