Python-PrepTerm
Quiz

Code:	MT2020509

1. rrect way to draw a line in canvas tkinter?
    1. line ()
    2. canvas. create_line ()
    3. create_line (canvas)
    4. None of the above
2. Which of the following environment variable for Python is an alternative module search path?
    1. PYTHONPATH
    2. PYTHONSTARTUP
    3. PYTHONCASEOK
    4. PYTHONHOME
3. 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)
4. 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**)
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. Pylab is a package that combine _____, _____ and _____ into a single namespace.

    1. Numpy, scipy and matplotlib
    2. Numpy, matplotlib and pandas
    3. Numpy, pandas and matplotlib
    4. Numpy, scipy and pandas

7. What will be the output of the code?

    z = "Best website is Tutorials Point" z.find("Tutorials")

    1. 3
    2. 13
    3. 17
    4. 16

8. 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.

9. 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

10. 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

11. 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

12. 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)

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. Essential thing to create a window screen using tkinter Python?

    1. call tk() function
    2. create a button
    3. To define a geometry
    4. All of the above

15. 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)

16. 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.

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 is output for **min**(`"hello world"`)

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

19. 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

20. 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)