Part I

1. What is printed by the Python code?

```
x = 5
y = x + 3
x = x - 1
z = 10
x = x + z
print('x: {}, y: {}, z: {}'.format(x, y, z))
```

2. What is printed by the Python code?

```
print(14//4, 14%4, 14.0/4)
```

3. What is printed by the Python code?

```
print(2*'No' + 3*'!')
print(2 * ('No' + 3*'!'))
```

4. What is printed by the Python code? Be careful: Note the backslashes:

```
print('how\nis it\nnow')
```

5. What is printed by the Python code?

```
for z in [2, 4, 7, 9]:
    print(z - 1)
```

6. What is printed by the Python code?

```
print('2' + '3')
```

7. What is printed by the Python code?

```
def f1():
    print('Hi')
def f2():
    print('Lo')
f2()
f1()
f1()
```

8. What is printed by the Python code?

```
def func():
     print('Yes')
print('No')
func()
```

9. What is printed by the Python code?

```
def func(x):
     print(2*x)
func(5)
func(4)
```

10. What is printed by the Python code?

```
def func(x):
     return x - 1
print(func(3) * func(5))
```

11. Which one of the following if statements will not execute successfully:

```
a. if (1, 2):

        print('foo')
b. if (1, 2):
                            print('foo')
c. if (1, 2): print('foo')
d. if (1, 2):
     print('foo')
e. if (1, 2):
          print('foo')
```

12. What is output of following code?

```
l = [1,2,6,5,7,8]
l.insert(9)
```

13. What will be the value of x?

```
x = ~~~~19
print(x)
```

14. What is output of following code?

```
num=10
```

```
while True:
     if (num%3 == 0):
          Break
     print(num)
     num += 1
```

15. What is the output of `['Hi!'] * 4`?

16. What is output for?

```
2 * 2 **3
```

17. What is printed by the Python code?

```
n = 3
for x in [2, 5, 8]:
     n = n + x
print(n)
```

18. What is printed by the Python code?

```
print(list(range(3)))
```

19. What is printed by the Python code?

```
for i in range(3):
     print('Hello again!')
```

20. What is printed by the Python code?

```
for i in range(4):
     print(i)
```

21. What is printed by the Python code?

```
def s(x): #1
     return x*x #2
for n in [1, 2, 10]: #3
     print(s(n)) #4
```

22. What is printed by the Python code?

```
def s(x): #1
     return x*x #2
```

```
tot = 0 #3
for n in [1, 3, 5]: #4
     tot = tot + s(n) #5
print(tot) #6
```

23. What is printed by the Python code?

```
x = 2.5679
y = 9.0
print('Answers {:.3f} and {:.3f}'.format(x, y))
```

24. What is printed by the Python code? d = dict() d['left'] = '<<' d['right'] = '>>' print('{left} and {right} or {right} and {left}'.format(**d))

```
<< and >> or >> and << Formats with {key} substitute strings
from the dictionary
```

25. Write a Python program that prompts the user for two numbers, reads them in, and prints out the product, labeled.

```
x = int(input('Enter a number: ')) # or some such prompt y =
int(input(”Enter another number: ”)) # or some such prompt
print('The product is ', x*y) # or some such label
```

26. Given a string s, write a short expression for a string that includes s repeated five times.

```
s*5 # or: s+s+s+s+s
```

27. Suppose you know x is an integer and ys is a string representing an integer. For instance, x is 3 and ys is '24'. Write code to print out the arithmetic sum of the two. In the example case, 27 would be printed.

```
print(x + int(ys))
```

28. Suppose you are given a list of words, wordList. Write Python code that will write one line for each word, repeating that word twice. For example if wordList is ['Jose', 'Sue', 'Ivan'], then your code would print Jose Jose Sue Sue Ivan Ivan

```
for word in wordlist: # variable word is arbitrary
     print(word, word) # but must match here!
```

29. Write code to create a Python dictionary (the dict type). Add two entries to the dictionary: Associate the key ‘name’ with the value ‘Juan’, and associate the key ‘phone’ with ‘508-1234’

```
d = dict() # name d is arbitrary, but match it in the next
lines d['name'] = 'Juan' d['phone'] = '508-1234'
```

30. Complete the code for the following function so it matches its documentation: def doubleList(numberList): ''' For each of the numbers in the list numberList, print a line containing twice the original number. For example, doubleList([3, 1, 5]) would print 6 2 10 '''

    ```
    def doubleList(numberList): ''' skip repeating docs... ''' for
    n in numberList: print(2*n)
    ```

31. Assuming a function process is already defined, write two lines of code, using a forloop, that is equivalent to the following: process('Joe') process('Sue') process('Ann') process('Yan')

    ```
    for name in ['Joe', 'Sue', 'Ann', 'Yan']: process(name)
    ```

32. Complete the function definition so it returns the square of the product of the parameters, so sqrProd(2, 5) returns (2*5)*(2*5) = 100. def sqrProd(x, y):

    ```
    return x*x*y*y # or: return (x*y)**2
    ```

33. What is the out of the code?

    ```
    def rev_func(x,length):
       print(x[length-1],end='' '')
       rev_func(x,length-1)
    x=[11, 12, 13, 14, 15]
    rev_func(x,5)
    ```

    a. The program runs fine without error.
    b. Program displays 15 14 13 12 11.
    c. Program displays 11 12 13 14 15.
    d. Program displays 15 14 13 12 11 and then raises an index out of range exception.

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

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

    ```
    a. <class 'float'>
    b. <class 'int'>
    c. NameError: '½' is not defined.
    ```

d. 0.5

35. Assume the following list definition in Python.
```
>>> letters = ["a", "b", "o", "c", "p"]
```
What would be displayed in a Python shell for each of the following expressions if they are evaluated in the given order? If it would give an error then write error.
```
>>> letters[1]
________________
>>> letters[len(letters)-2]
_________________
>>> letters + ["x"]
__________________
>>> letters
__________________
```

36. Show how to create a list of every integer between 0 and 100 , inclusive, named nums1 using Python, sorted in increasing order.
```
nums1 = list(_______________________________________)
```

37. Let `nums2` and `nums3` be two non-empty lists. Write a Python command that will append the last element of `nums3` to the end of `nums2` .
```
________________.append(______________________).
```

38. In economics, the percentage rate of inflation for a period of time is calculated based on the final value F of a commodity and the initial value I of the commodity, using the formula ((F −I)/I)×100. Write a Python function inflation rate(initial, final) to compute and return the inflation rate given the initial and final values of a commodity.

```
def inflation_rate(initial,final):
     return ((final - initial)/ initial) * 100
```

39. Using the function from the previous question, write a Python function `average_inflation_rate()`, that computes and returns the average rate of inflation for the 3-year period represented in the table below:

Year	Initial value	Final value
1	23.50	24.00
2	24.00	24.25
3	24.25	24.38

The function is required to call the function from the previous question.

```
def average_inflation_rate():
     year1 = inflation_rate(23.50,24.00)
     year2 = inflation_rate(24.00,24.25)
```

```
        year3 = inflation_rate(24.25,24.38)
        return (year1 + year2 + year3)/3
```

40. Consider the simple function given below.

```
def twice(n):
    print(2*n)
```

When we compare `twice(5)` with 10 the Python interpreter would return False after printing 10 as shown below. Explain why it gives False as a result of the comparison.

```
>>> twice(5) == 10
10
False
```

Since 2∗n is printed, not returned, twice(5) would return None. Comparing a None value to an integer would yield False.

41. Consider the following Python function where m and n are assumed to be a positive integers:

```
def mystery(n, m):
    p = 0
    e = 0
    while e < m:
        p = p + n
        e = e + 1
    return p
```

Trace this function for n = 4, m = 3, showing the value of e and p in the table above at the end of each iteration of the loop. The initial values of p and e are given for you in the table. Use as many spaces as you need.

```
p       e
=======
0       0
4       1
8       2
12      3
```

42. Which of the following functions is being computed by mystery above? Circle your answer.
    a. nm
    b. n + m
    c. n
    d. m
    e. mn

f. None of these

43. Suppose that the return statement was indented as below. What would mystery(4, 3) return in this case?
```
def mystery(n, m):
     p = 0
     e = 0
     while e < m:
          p = p + n
          e = e + 1
          return p
```

44. _____ represents an entity in the real world with its identity and behaviour.
   a. A method
   b. An object
   c. A class
   d. An operator

45. _____ is used to create an object.
   a. Class
   b. constructor
   c. User-defined functions
   d. In-built functions

46. What will be the output of the following Python code?

```
class test:
     def __init__(self,a="Hello World"):
         self.a=a

     def display(self):
         print(self.a)
obj=test()
obj.display()
```

   a. The program has an error because constructor can't have default arguments
   b. Nothing is displayed
   c. "Hello World" is displayed
   d. The program has an error display function doesn't have parameters

47. What is `setattr()` used for?
   a. To access the attribute of the object
   b. To set an attribute
   c. To check if an attribute exists or not

d. To delete an attribute

48. What is `getattr()` used for?
    a. To access the attribute of the object
    b. To delete an attribute
    c. To check if an attribute exists or not
    d. To set an attribute

49. What will be the output of the following Python code?

```
class change:
    def __init__(self, x, y, z):
        self.a = x + y + z

x = change(1,2,3)
y = getattr(x, 'a')
setattr(x, 'a', y+1)
print(x.a)
```

    a. 6
    b. 7
    c. Error
    d. 0

50. What will be the output of the following Python code?

```
class test:
     def __init__(self,a):
         self.a=a

     def display(self):
         print(self.a)
obj=test()
obj.display()
```

    a. Runs normally, doesn’t display anything
    b. Displays 0, which is the automatic default value
    c. Error as one argument is required while creating the object
    d. Error as display function requires additional argument

51. Is the following Python code correct?

```
>>> class A:
     def __init__(self,b):
            self.b=b
        def display(self):
            print(self.b)
>>> obj=A("Hello")
>>> del obj
```

a. True
b. False

52. What will be the output of the following Python code?

```
class test:
    def __init__(self):
        self.variable = 'Old'
        self.Change(self.variable)
    def Change(self, var):
        var = 'New'
obj=test()
print(obj.variable)
```

a. Error because function change can't be called in the __init__ function
b. 'New' is printed
c. 'Old' is printed
d. Nothing is printed

53. What is Instantiation in terms of OOP terminology?
a. Deleting an instance of class
b. Modifying an instance of class
c. Copying an instance of class
d. Creating an instance of class

54. What will be the output of the following Python code?

```
class fruits:
    def __init__(self, price):
        self.price = price
obj=fruits(50)

obj.quantity=10
obj.bags=2

print(obj.quantity+len(obj.__dict__))
```

a. 12

b. 52
c. 13
d. 60

55. What will be the output of the following Python code?

```
 class Demo:
     def __init__(self):
         pass

     def test(self):
         print(__name__)

obj = Demo()
obj.test()
```

a. Exception is thrown
b. __main__
c. Demo
d. Test

56. The assignment of more than one function to a particular operator is _______
a. Operator over-assignment
b. Operator overriding
c. Operator overloading
d. Operator instance

57. Which of the following is not a class method?
a. Non-static
b. Static
c. Bounded
d. Unbounded

58. What will be the output of the following Python code?

```
def add(c,k):
    c.test=c.test+1
    k=k+1
class A:
    def __init__(self):
        self.test = 0
def main():
    Count=A()
    k=0
    for i in range(0,25):
        add(Count,k)
    print("Count.test=", Count.test)
    print("k =", k)
main()
```

```
   a. Exception is thrown
   b.
      Count.test=25
      k=25
   c.
      Count.test=25
      k=0
   d.
      Count.test=0
      k=0
```

59. Which of the following Python code creates an empty class?

   a.
   ```
   class A:
     Return
   ```
   b.
   ```
   class A:
     Pass
   ```
   c.
   ```
   class A:
   ```
   d.
   It is not possible to create an empty class

60. Is the following Python code valid?

```
class B(object):
  def first(self):
    print("First method called")
  def second():
    print("Second method called")
ob = B()
B.first(ob)
```

   a. It isn't as the object declaration isn't right
   b. It isn't as there isn't any __init__ method for initializing class members
   c. Yes, this method of calling is called unbounded method call
   d. Yes, this method of calling is called bounded method call

61. What are the methods which begin and end with two underscore characters called?
    a. Special methods
    b. In-built methods
    c. User-defined methods
    d. Additional methods

62. Special methods need to be explicitly called during object creation.
    a. True
    b. False

63. What will be the output of the following Python code?

```
>>> class demo():
     def __repr__(self):
          return '__repr__ built-in function called'
     def __str__(self):
          return '__str__ built-in function called'
>>> s=demo()
>>> print(s)
```

    a. Error
    b. Nothing is printed
    c. __str__ called
    d. __repr__ called

64. What is hasattr(obj,name) used for?
    a. To access the attribute of the object
    b. To delete an attribute
    c. To check if an attribute exists or not
    d. To set an attribute

65. What will be the output of the following Python code?

```
class stud:
   def __init__(self, roll_no, grade):
      self.roll_no = roll_no
      self.grade = grade
   def display (self):
      print("Roll no : ", self.roll_no,  ", Grade: ",
self.grade)
stud1 = stud(34, 'S')
stud1.age=7
print(hasattr(stud1, 'age'))
```

a. Error as age isn’t defined
b. True
c. False
d. 7

66. What is delattr(obj,name) used for?
   a. To print deleted attribute
   b. To delete an attribute
   c. To check if an attribute is deleted or not
   d. To set an attribute

67. __del__ method is used to destroy instances of a class.
   a. True
   b. False

68. What will be the output of the following Python code?

```
class stud:
   ‘Base class for all students’
   def __init__(self, roll_no, grade):
      self.roll_no = roll_no
      self.grade = grade
   def display (self):
      print("Roll no : ", self.roll_no,  ", Grade: ", 
self.grade)
print(student.__doc__)
```

   a. Exception is thrown
   b. __main__
   c. Nothing is displayed
   d. Base class for all students

69. What does print(Test.__name__) display (assuming Test is the name of the class)?
   a. ()
   b. Exception is thrown
   c. Test
   d. __main__

70. Which of the following best describes inheritance?
   a. Ability of a class to derive members of another class as a part of its own definition
   b. Means of bundling instance variables and methods in order to restrict access to certain class members
   c. Focuses on variables and passing of variables to functions

   d. Allows for implementation of elegant software that is well designed and easily modified

71. Which of the following statements is wrong about inheritance?
   a. Protected members of a class can be inherited
   b. The inheriting class is called a subclass
   c. Private members of a class can be inherited and accessed
   d. Inheritance is one of the features of OOP

72. What will be the output of the following Python code?

```
class Demo:
    def __new__(self):
        self.__init__(self)
        print("Demo's __new__() invoked")
    def __init__(self):
        print("Demo's __init__() invoked")
class Derived_Demo(Demo):
    def __new__(self):
        print("Derived_Demo's __new__() invoked")
    def __init__(self):
        print("Derived_Demo's __init__() invoked")
def main():
    obj1 = Derived_Demo()
    obj2 = Demo()
main()
```

   a.
      Derived_Demo’s __init__() invoked
      Derived_Demo's __new__() invoked
      Demo's __init__() invoked
      Demo's __new__() invoked
   b.
      Derived_Demo's __new__() invoked
      Demo's __init__() invoked
      Demo's __new__() invoked
   c.
      Derived_Demo's __new__() invoked
      Demo's __new__() invoked
   d.
      Derived_Demo’s __init__() invoked
      Demo's __init__() invoked

73. What will be the output of the following Python code?

```
class Test:
    def __init__(self):
        self.x = 0
class Derived_Test(Test):
    def __init__(self):
        self.y = 1
def main():
    b = Derived_Test()
    print(b.x,b.y)
main()
```

a. 0 1
b. 0 0
c. Error because class B inherits A but variable x isn't inherited
d. Error because when object is created, argument must be passed like Derived_Test(1)

74. What will be the output of the following Python code?

```
class A():
    def disp(self):
        print("A disp()")
class B(A):
    pass
obj = B()
obj.disp()
```

a. Invalid syntax for inheritance
b. Error because when object is created, argument must be passed
c. Nothing is printed
d. A disp()

75. All subclasses are a subtype in object-oriented programming.
    a. True
    b. False

76. When defining a subclass in Python that is meant to serve as a subtype, the subtype Python keyword is used.
    a. True
    b. False

77. Suppose B is a subclass of A, to invoke the __init__ method in A from B, what is the line of code you should write?

a. A.__init__(self)
b. B.__init__(self)
c. A.__init__(B)
d. B.__init__(A)

78. What will be the output of the following Python code?

```
class Test:
    def __init__(self):
        self.x = 0
class Derived_Test(Test):
    def __init__(self):
        Test.__init__(self)
        self.y = 1
def main():
    b = Derived_Test()
    print(b.x,b.y)
main()
```

a. Error because class B inherits A but variable x isn’t inherited
b. 0 0
c. 0 1
d. Error, the syntax of the invoking method is wrong

79. What will be the output of the following Python code?

```
class A:
    def __init__(self, x= 1):
        self.x = x
class der(A):
    def __init__(self,y = 2):
        super().__init__()
        self.y = y
def main():
    obj = der()
    print(obj.x, obj.y)
main()
```

a. Error, the syntax of the invoking method is wrong
b. The program runs fine but nothing is printed
c. 1 0
d. 1 2

80. What does built-in function type do in context of classes?
a. Determines the object name of any value

b. Determines the class name of any value
c. Determines class description of any value
d. Determines the file name of any value

81. What will be the output of the following Python code?

```
class A:
    def one(self):
        return self.two()

    def two(self):
        return 'A'

class B(A):
    def two(self):
        return 'B'
obj1=A()
obj2=B()
print(obj1.two(),obj2.two())
```

a. A A
b. A B
c. B B
d. An exception is thrown

82. What will be the output of the following Python code?

```
class A:
     def __init__(self):
         self.__i = 1
         self.j = 5

     def display(self):
         print(self.__i, self.j)
class B(A):
     def __init__(self):
         super().__init__()
         self.__i = 2
         self.j = 7
c = B()
c.display()
```

a. 2 7
b. 1 5
c. 1 7

d. 2 5

83. Which of the following statements isn’t true?
    a. A non-private method in a superclass can be overridden
    b. A derived class is a subset of superclass
    c. The value of a private variable in the superclass can be changed in the subclass
    d. When invoking the constructor from a subclass, the constructor of superclass is automatically invoked

84. What will be the output of the following Python code?

```
class A:
    def __init__(self,x):
        self.x = x
    def count(self,x):
        self.x = self.x+1
class B(A):
    def __init__(self, y=0):
        A.__init__(self, 3)
        self.y = y
    def count(self):
        self.y += 1
def main():
    obj = B()
    obj.count()
    print(obj.x, obj.y)
main()
```

    a. 3 0
    b. 3 1
    c. 0 1
    d. An exception in thrown

85. What will be the output of the following Python code?

```
>>> class A:
     pass
>>> class B(A):
     pass
>>> obj=B()
>>> isinstance(obj,A)
```

    a. True
    b. False
    c. Wrong syntax for isinstance() method

d. Invalid method for classes

86. What will be the output of the following Python code?

```
class A:
    def test1(self):
        print(" test of A called ")
class B(A):
    def test(self):
        print(" test of B called ")
class C(A):
    def test(self):
        print(" test of C called ")
class D(B,C):
    def test2(self):
        print(" test of D called ")
obj=D()
obj.test()
```

a.
   test of B called
   test of C called
b.
   test of C called
   test of B called
c. test of B called
d. Error, both the classes from which D derives has same method test()

87. What will be the output of the following Python code?

```
class A:
    def test(self):
        print("test of A called")
class B(A):
    def test(self):
        print("test of B called")
        super().test()
class C(A):
    def test(self):
        print("test of C called")
        super().test()
class D(B,C):
    def test2(self):
        print("test of D called")
obj=D()
```

```
obj.test()
```

a.
    test of B called
    test of C called
    test of A called

b.
    test of C called
    Test of B called

c.
    test of B called
    test of C called

d. Error, all the three classes from which D derives has same method test()

88. What will be the output of the following Python code?

```
x = ['ab', 'cd']
for i in x:
    i.upper()
print(x)
```

a. [‘ab’, ‘cd’]
b. [‘AB’, ‘CD’]
c. [None, None]
d. none of the mentioned

89. What will be the output of the following Python code?

```
x = ['ab', 'cd']
for i in x:
    x.append(i.upper())
print(x)
```

a. [‘AB’, ‘CD’]
b. [‘ab’, ‘cd’, ‘AB’, ‘CD’]
c. [‘ab’, ‘cd’]
d. none of the mentioned

90. What will be the output of the following Python code?

```
i = 1
while True:
    if i%3 == 0:
        break
    print(i)
    i + = 1
```

a. 1 2
b. 1 2 3
c. error
d. none of the mentioned

91. What will be the output of the following Python code?

```
i = 1
while True:
    if i%7 == 0:
        break
    print(i)
    i += 1
```

a. 1 2 3 4 5 6
b. 1 2 3 4 5 6 7
c. error
d. none of the mentioned

92. What will be the output of the following Python code?

```
i = 5
while True:
    if i%11 == 0:
        break
    print(i)
    i += 1
```

a. 5 6 7 8 9 10
b. 5 6 7 8
c. 5 6
d. error

93. 6. What will be the output of the following Python code?

```
i = 5
while True:
    if i%9 == 0:
        break
    print(i)
    i += 1
```

a. 5 6 7 8
b. 5 6 7 8 9
c. 5 6 7 8 9 10 11 12 13 14 15 ….
d. error

94. What will be the output of the following Python code?

```
i = 1
while True:
    if i%2 == 0:
        break
    print(i)
    i += 2
```

a. 1
b. 1 2
c. 1 2 3 4 5 6 …
d. 1 3 5 7 9 11 …

95. What will be the output of the following Python code?

```
i = 2
while True:
    if i%3 == 0:
        break
    print(i)
    i += 2
```

a. 2 4 6 8 10 …
b. 2 4
c. 2 3
d. error

96. What will be the output of the following Python code?

```
i = 1
while False:
    if i%2 == 0:
        break
    print(i)
    i += 2
```

a. 1

b. 1 3 5 7 …
c. 1 2 3 4 …
d. none of the mentioned

97. What will be the output of the following Python code?

```
True = False
while True:
    print(True)
    break
```

a. True
b. False
c. None
d. none of the mentioned

98. What is the type of each element in sys.argv?
a. set
b. list
c. tuple
d. string

99. What is the length of sys.argv?
a. number of arguments
b. number of arguments + 1
c. number of arguments – 1
d. none of the mentioned

100. What will be the output of the following Python code?

```
def foo(k):
    k[0] = 1
q = [0]
foo(q)
print(q)
```

a. [0]
b. [1]
c. [1, 0]
d. [0, 1]

101. How are keyword arguments specified in the function heading?
a. one-star followed by a valid identifier
b. one underscore followed by a valid identifier

   c. two stars followed by a valid identifier
   d. two underscores followed by a valid identifier

102. How many keyword arguments can be passed to a function in a single function call?
   a. zero
   b. one
   c. zero or more
   d. one or more

103. What will be the output of the following Python code?

```
def foo(fname, val):
    print(fname(val))
foo(max, [1, 2, 3])
foo(min, [1, 2, 3])
```

   a. 3 1
   b. 1 3
   c. error
   d. none of the mentioned

104. What will be the output of the following Python code?

```
def foo():
    return total + 1
total = 0
print(foo())
```

   a. 0
   b. 1
   c. error
   d. none of the mentioned

105. What will be the output of the following Python code?

```
def foo():
    total += 1
    return total
total = 0
print(foo())
```

   a. 0
   b. 1
   c. error

d. none of the mentioned

106. What will be the output of the following Python code?

```
def foo(x):
    x = ['def', 'abc']
    return id(x)
q = ['abc', 'def']
print(id(q) == foo(q))
```

a. True
b. False
c. None
d. Error

107. What will be the output of the following Python code?

```
def foo(i, x=[]):
    x.append(i)
    return x
for i in range(3):
    print(foo(i))
```

a. [0] [1] [2]
b. [0] [0, 1] [0, 1, 2]
c. [1] [2] [3]
d. [1] [1, 2] [1, 2, 3]

Part II

1. What does DBMS stand for?
    a. Database Management Software
    b. Data Binary Management System
    c. Database Management System
    d. Data Batch Management Software

2. Which statement correctly inserts a row into a table named 'cars'?
    a. ADD INTO cars VALUES ('1', 'Toyota', 'Corolla');
    b. INSERT INTO cars VALUES ('1', 'Toyota', 'Corolla');
    c. INSERT cars ('1', 'Toyota', 'Corolla');
    d. UPDATE cars SET VALUES ('1', 'Toyota', 'Corolla');

3. What is the primary function of the SQL SELECT statement?
    a. To update the information in a database
    b. To insert new data into a table
    c. To retrieve data from a database
    d. To delete data from a table

4. Which of the following is NOT a component of a database management system as described in the slides?
    a. Files
    b. Tables
    c. Rows
    d. Columns

5. What is the purpose of the 'WHERE' clause in SQL?
    a. To specify which database to use
    b. To set the table for data insertion
    c. To define conditions for selecting, updating, or deleting data
    d. To list the databases available

6. What is the first step to use MySQL in a Python script?
    a. Write a SQL query
    b. Install the mysql.connector module
    c. Create a database
    d. Generate a cursor object

7. How do you test if your MySQL connector is installed correctly in Python?
    a. Run a complex SQL query

    b. Check the version of the MySQL connector
    c. Import mysql.connector in your script
    d. Connect to the MySQL database without username and password

8. What Python object is primarily used to execute SQL commands?
    a. Database
    b. Connector
    c. Cursor
    d. Script

9. Which command in Python checks the established connection to the MySQL database?
    a. db.status()
    b. db.test()
    c. print(db)
    d. db.connect()

10. In the context of this lab, what does 'carmax' refer to?
    a. A Python module
    b. A variable in a script
    c. A database
    d. A table in the database

11. Which Python framework is built into the Python standard library for developing GUI applications?
    a. PyQt
    b. WxPython
    c. Tkinter
    d. Flask

12. What is the primary function of the 'mainloop()' method in a Tkinter application?
    a. To open a new window
    b. To close the application
    c. To listen for events and keep the application running
    d. To update the user interface

13. Which widget in Tkinter is used to display simple text to a user?
    a. Button
    b. Label
    c. Entry
    d. Frame

14. How do you set the size of a Tkinter window to 500x100 pixels?
    a. w.size("500x100")
    b. w.set("500x100")

c. w.resize(500, 100)
d. w.geometry("500x100")

15. What method is used to place a widget at a specific position inside a Tkinter window?
    a. pack()
    b. place()
    c. set()
    d. put()

16. What is the purpose of the Canvas widget in tkinter?
    a. To display text in multiple fonts and sizes.
    b. To organize other widgets into a grid.
    c. To provide a rectangular area intended for drawing pictures or other complex layouts.
    d. To create a menu system for applications.

17. Which method is used to create an arc on a Canvas in tkinter?
    a. create_line()
    b. create_arc()
    c. create_image()
    d. create_polygon()

18. How can you add an image to a Canvas in tkinter?
    a. Using the create_text() method with an image file as the text.
    b. By setting the background property of the Canvas to an image file.
    c. Using the create_image() method with a PhotoImage or BitmapImage.
    d. By importing an image directly onto the Canvas without any methods.

19. What does the pack() method do in the context of a tkinter Canvas?
    a. It compresses the image files to save space.
    b. It sends the Canvas widget to the printer.
    c. It organizes widgets in a tabular form.
    d. It adds the Canvas widget to the window and displays it.

20. Which tkinter widget is used to group and organize other widgets in a user-friendly way?
    a. Label
    b. Frame
    c. Canvas
    d. Button

21. What is Flask primarily used for?
    a. Writing low-level networking programs.
    b. Creating standalone WSGI containers.
    c. Developing web applications.

    d. Managing databases.

22. Which of the following servers is recommended for running Flask applications in production?
    a. Flask's built-in server.
    b. Apache.
    c. Gunicorn.
    d. Nginx.

23. How can you secure a Flask application using HTTPS?
    a. By using the secure Flask extension.
    b. By configuring Flask to use SSL certificates.
    c. By redirecting all traffic through HTTPS using a proxy server.
    d. Flask applications are secure by default.

24. What does the route() decorator in Flask do?
    a. It optimizes the Flask application.
    b. It binds a function to a URL.
    c. It redirects the user to a different URL.
    d. It performs URL masking.

25. Which function in Flask is used to generate URLs for a function?
    a. generate_url()
    b. url_for()
    c. link_to()
    d. bind_url()

26. What is the primary use of the render_template() function in Flask?
    a. To process form data.
    b. To render an HTML page using templates.
    c. To configure the Flask application.
    d. To send JSON data.

27. What is the primary benefit of using threads in a program?
    a. To increase the processing power of the CPU.
    b. To perform multiple tasks simultaneously within a program.
    c. To enhance the security of the program.
    d. To allocate more memory to a program.

28. Which method starts a thread in Python?
    a. begin()
    b. init()
    c. run()
    d. start()

29. What does the join() method do in threading?
    a. It combines two threads into one.
    b. It pauses the execution of the thread.
    c. It forces a thread to wait for another thread to finish.
    d. It checks if the thread is completed.

30. What are sockets used for in programming?
    a. To create graphical user interfaces.
    b. To manage databases.
    c. To handle bidirectional communications between processes.
    d. To generate and manage threads.

31. What method is used to listen for incoming connections in socket programming?
    a. connect()
    b. bind()
    c. listen()
    d. accept()

32. Which method retrieves a client connection from the server socket?
    a. open()
    b. receive()
    c. accept()
    d. connect()

33. What is the purpose of an inner class in Python?
    a. To create independent classes with no relations to other classes.
    b. To encapsulate logic and data related to another class within it.
    c. To improve the speed of code execution.
    d. To create multiple instances of the main class.

34. What is an iterator in Python?
    a. A data type that stores multiple items.
    b. A class that contains only private methods.
    c. An object that enables traversal through a container, particularly for loops.
    d. A tool used for network programming.

35. Which keyword is used to create a generator in Python?
    a. iterate
    b. generate
    c. yield
    d. Return

36. What is a DataFrame in Pandas?

a. A function to execute mathematical operations.
b. A data structure for storing data in tabular form.
c. A method for plotting data charts.
d. A type of database used in Python.

37. Which function is used to read a CSV file into a Pandas DataFrame?
a. pandas.read_csv()
b. pandas.open()
c. pandas.import_data()
d. pandas.load_csv()

38. How can you select rows in a DataFrame based on a condition?
a. DataFrame.filter(condition)
b. DataFrame.where(condition)
c. DataFrame[DataFrame[column] > value]
d. DataFrame.select(condition)

39. What is the primary use of Matplotlib in Python?
a. Data cleaning and transformation.
b. Web development.
c. Creating 2D plots and visualizations.
d. Managing databases.

40. Which function in Matplotlib is used to create a line plot?
a. plt.line()
b. plt.plot()
c. plt.graph()
d. plt.show()

41. How can you add a title to a plot in Matplotlib?
a. plt.title('Title Name')
b. plt.add_title('Title Name')
c. plt.set_title('Title Name')
d. plt.label('Title Name')