1. What Is Object-Oriented Programming?

In [ ]:

```
The object oreinted programming is based on the class and object.
-The classes are are made from thye data. and are blueprint of objects
-the object contains variables and methods to operate on the data.
object describes the class..
```

2. Difference between Procedural programming and OOPs?

In [ ]:

```
-difference between pop and oop
        pop                                                    oop
    1. in pop program is divided into small          1.program is divided into small par
    parts called fumctions.

    2.it folllows top to down approach               2. it follows bottom to up programm

    3. does not allow data hiding in proper way.     3.provides data hiding, so is more
     hence it is less secure.

    4.here function is more important                4. here data is important.

    5.based on unreal world                          5. based on real world
```

3. What are the fundamental principles/features of Object-Oriented Programming?

In [ ]:

```
-Features of oops:
    1. inheritance
    2. polymorphism
    3. absraction
    4. encapsulation
```

4. What is an object?

In [ ]:

```
-Object is a real time entity which has behavior and states.
-e.g. dog has states asn like name, clor etc. and behavour like barking
```

5. What is a class?

In [ ]:

```
class is a bluprint of object. it defines the behavior and functionality of object.
```

6. What is the difference between a class and an object?

In [ ]:

```
-difference between class and object
                    class                                                  object
1. memory is not allocated                                        1. memory is allocated

2. class is created only once                                     2. object is created more t

3.class cannot be manipulated                                     3.object can be manipulated

4.class is used to bind the data and methods                      4. object is like a variabl
```

7. Can you call the base class method without creating an instance?

In [ ]:

```
yes. a base class method can be called wityhot using instance.
the way is by using inheritance.
in inheritace other class which can be called child class inherits all the variable and met
and can be called using instance of child class.
```

8. What is inheritance?

In [ ]:

```
Inheritance:
    - inheritance is a method which allows a class to inherit the properties of another cla
    - the class inheriting prperties is child or derived class.
    - class from which properties are inherited is base class or parent class
```

9. What are the different types of inheritance?

```
There are four types of inheritance:
    1.single inheritance:
        in single inheritance, one base class properties are derived by only one derived
class.
                                1 base class/parent
                                       ||
```

```
                              1 derived class/child
    2. multiple interitance:
        in multiple inheritance, there is one base class and multiple derived classes.
        hence many classes inherit the properties of only one class.
                              base class/parent class
                                 //      ||      \\
                               child1  child2   child3

    3. multilevel inheritance:
         in this there is a parent, child and grandchild relationship in classes.
         that is each class derives some properties from its bas class.

                              parent class
                                   ||
                               child class
                                   ||
                              grandchild class
```

In [ ]:

```
10. What is the difference between multiple and multilevel inheritances?
11. What are the limitations of inheritance?
12. What are the superclass and subclass?
13. What is the super keyword?
14. What is encapsulation?

16. What is the difference between public and private access modifiers?
17. Is Python 100 percent object-oriented?
18. What is data abstraction?
19. How to achieve data abstraction?
20. What is an abstract class?
21. Can you create an object of an abstract class?
22. Differentiate between data abstraction and encapsulation
23. What is polymorphism?
24. What is the overloading method?
25. What are the limitations of OOPs?
```

10. What is the difference between multiple and multilevel inheritances?

```
difference between multiple and multilevel inheritance:
                   multiple inheritance                                    multilevel
inheritance
        1. Multiple Inheritance is an Inheritance type              1.Multilevel
Inheritance is an Inheritance type that
           where a class inherits from more than one base class.       inherits from a
derived class, making that
                                                                       derived class a
base class for a new class.

        2.Multiple Inheritance is not widely used because            2. Multilevel
Inheritance is widely used.
          it makes the system more complex.

        3. Multiple Inheritance has two class levels namely,         3. Multilevel
Inheritance has three class levels namely,
```

```
                                                                    base class,
intermediate class and derived class.
            base class and derived class.
```

11. What are the limitations of inheritance?

```
-Inherited functions work slower than normal function as there is indirection.
-Improper use of inheritance may lead to wrong solutions.
-Often, data members in the base class are left unused which may lead to memory wastage.
-Inheritance increases the coupling between base class and derived class.
```

12. What are the superclass and subclass?

superclass: In inheritance, one class can inherit the properties of other class. the class from which properties are inherited, is superclass

subclass: the subclass is class inheriting the properties.

13. What is the super keyword?

The super keyword refers to superclass (parent) objects. It is used to call superclass methods, and to access the superclass constructor. The most common use of the super keyword is to eliminate the confusion between superclasses and subclasses that have methods with the same name.

14. What is encapsulation?

Encapsulation: it refers to bundling of data into the methods or functions to which it belongs. - due to this only authorized entity can access that data. -encapsulation is done by using private variables and methods - it hides the data i.e. method and variables - the private variables can be defined using (__) double undersore. e.g. __add() # private variable

15. What is the name mangling and how does it work?

```
# -Name mangling is the loophole in encasulation.
-the variables and method declared as private to hide the data can be accessed through
 name mangling process.
- in name mangling, the private variables can be accessed using _ preceding to classname.
eg.
     class Employee():
        def _init_(self,name):
            self.name=name
        def emp_det(self,empname):
            __empname="sagar"
            print("empname", __empname)
    obj=Employee()
    obj._Employee.__empname()
```

```
    >> o/p >> "sagar"
```

16. What is the difference between public and private access modifiers?

```
#                       public modifier                                   private modifier
    1. This modifier is applicable for both top-level          1. This modifier is not
applicable for both top-level
                                                                  classes and interfaces.
       classes and interfaces

    2. Public members can be accessed from the child           2. Private members cannot
be accessed from the child
                                                                  class of the same
package.
       class of the same package.

    3. Public member can be accessed from non-child            3. Private members cannot
be accessed from non-child class
                                                                   of same package.
       class of same package.
    4. Public members can be accessed from child               4. Private members cannot
be accessed from child class
                                                                   of outside package.
       class of outside package.

    5. Public modifier is the most accessible modifier.         5. Private modifier is
the most restricted modifier.
```

17. Is Python 100 percent object-oriented?

In [ ]:

```
Python supports all the concept of "object oriented programming" but it is NOT fully object
because - The code in Python can also be written without creating classes.
```

18. What is data abstraction?

In [ ]:

```
-Data Abstraction is used to hide the internal functionality of the function from the users
-The users only interact with the basic implementation of the function, but inner working i
-using this user dont need to deal with the complexity of any code.
-he knows what the code does, but dont know how it does.
```

19. How to achieve data abstraction?

In [ ]:

```
-In Python, abstraction can be achieved by using abstract classes and interfaces.
-A class that consists of one or more abstract method is called the abstract class.
=Abstract methods do not contain their implementation.
= Python provides the abc module to use the abstraction in the Python program.
```

20. What is an abstract class?

In [ ]:

```
Abstract Class is a type of class in OOPs, that declare one or more abstract methods.
These classes can have abstract methods as well as concrete methods.
A normal class cannot have abstract methods. An abstract class is a class that contains at
```

21. Can you create an object of an abstract class

In [ ]:

```
no, object of abstract class cannot be created. its only for declaretion not for implementa
```

22. Differentiate between data abstraction and encapsulation

In [ ]:

```
                abstraction                                        encapsulation
   1. abstraction is pracees of hiding complicated code       1. encapsulation binds data
      from user.

   2. it makes easy to user to implement the function          2. it is use for securing
                                                                  unauthorised user.
      rather knowing the inner code

   3.it is achieved using abstract class and abc module        3. it is achievd using pri
                                                                  __priv_var
```

23. What is polymorphism?

In [ ]:

```
polymorphism is one of the features of oops.
poly means many, morhism means form.
so polymorphism is many forms of single function
single function can be used for mutiple type of data.
e.g.
    length()
    the len function can be used ti find the length of more than one data type.
    1. len("str")       2. len([list])    3. len(tuple)
```

24. What is the overloading method?

In [ ]:

```
Method overloading is a concept of Java in which we can create multiple methods of the same
and all methods work in different ways. When more than one method of the same name is creat
this type of method is called Overloaded Method.
it is same as polymorphism.
```

25. What are the limitations of OOPs?

In [ ]:

```
Size: OO programs are much larger than other programs. ...
Effort: OO programs require a lot of work to create. ...
Speed: OO programs are slower than other programs, partially because of their size.
```

In [ ]: