Roll_No._08_Urmila_kumari (Assignment_08)

In [ ]:
```
#1. What Is Object-Oriented Programming?
Whenever we use class and object then it's called OOPs.
```

In [ ]:
```
#2. Difference between Procedural programming and OOPs?
procedural program:
    1) It divides the program into small programs and refers to them as functions.
    2) It does not provides any inheritance.
    3) It priorities function over data.
    4) It is not very suitable for solving any big or complex problems.
    5) It is not very easy to add new functions and data in the procedural program.
Object Oriented Programming:
    1) It divides the program into small parts and refers to them as object.
    2) It achives inheritance in three modes- protected, private, and public.
    3) It prioritize data over function.
    4) It is suitable for solving any big or complex problems.
    5) It is very easy to add new functions and data in the Object oriented program.

```

In [ ]:
```
#3. What are the fundamental principles/features of Object-Oriented Programming?
There are four fundamental features of  Object-Oriented Programming.
1) Inheritance
2) Encapsulation
3) Polymerphism
4) Abstraction
```

In [ ]:
```
#4. What is an object?
1) It is the real world entities(House, Mobile, Birds etc.)
2) Instance of class
3) Everthing in Python is object.
   eg. str, list, tuple etc...

```

In [ ]:
```
#5. What is a class?
1) It is a blueprnt of an object that defineds its variables and methods.
2) It is the collection of objects.
```

In [ ]:
```
#6. What is the difference between a class and an object?
class:
    1) class is a logical entity.
    2) class is used as a templet for declearing and creating the objects.
    3) class has to be created first and only once.
    4) class can not be manupulated as they are not avilable in the memory.
object:
    1) Object is physical or real-world entity such as Mobile, house, book, car etc.
    2) An object is an instance of class.
    3) An object is created many times as per requirement.
    4) Object can be manupulated.
```

In [ ]:
```
#7. Can you call the base class method without creating an instance?
Yes it is posible.
1) If it is a static method.
2) From derived classes using base keyword.
```

In [ ]:
```
#8. What is inheritance?
Main aim of inheritance in OOPs is to reutilize the already existing code and build an advanced version of that.
we can call all methods and variables of parent call by creating the object of child class.
```

In [ ]:
```
#9. What are the different types of inheritance?
There are five types of inheritance :
    1) Single inheritance
    2) Multi level inheritance
    3) Multiple inheritance
    4) Heirachical inheritance
    5) Hybrid inheritance
```

In [ ]:
```
#10. What is the difference between multiple and multilevel inheritances?
Multi level inheritance:
    If we want to use class-1 properties in class-2 and class-2 properties in class-3 and
    class-3 properties in class-4 so on.
Multiple Inheritance:
    we can call multiple classes in child class.
```

In [ ]:
```
#11. What are the limitations of inheritance?
1) decreases the Execution sleed:
    loading multiple class because they are interdependent.
2) Tightly Coupled classes:
    this means that even though parent classes can be executed independently,
    child classes cannot be executed without defining their parent classes.
```

In [ ]:
```
#12. What are the superclass and subclass?
Base class/ parent class/ super class:
    The class from which a class inherits is called parentor super class.
subclass/ child class/ derived class:
    A class which inherits from a superclass is called a sub class.

```

In [ ]:
```
#13. What is the super keyword?
The super() function returns an object that represent the parent class.
```

In [ ]:
```
#14. What is encapsulation?
Used to restrict the access of variables and methods from outside the class.
Protecting the data from modification or to prevent accidental damages.
```

In [ ]:
```
#15. What is the name mangling and how does it work?
The name mangling process helps to access the class variables from outside the class.
```

In [ ]:
```
#16. What is the difference between public and private access modifiers?
Public access modifier:
    The public access modifier allows a code from outside or inside the class to access
    the class's methods and properties.
private access modifiers:
    private access modifiers prevents access to a class's method or properties
    from any code that is outside tha class.

```

In [ ]:
```
#17. Is Python 100 percent object-oriented?
Python supports all the concept of "Object Oriented Programming" but it is Not fully Object
oriented because -the code in Python can also be written without creating classes.
```

In [ ]:
```
#18. What is data abstraction?
Data abstraction is the reduction of a particular body of data to simplified representation of the whole.
basically, Abstraction focuses on hidding the internal implementation of process or method from the user.
```

In [ ]:
```
#19. How to achieve data abstraction?
In python, abstraction can be achived by using abstract classes and methods in our programs.
```

In [ ]:
```
#20. What is an abstract class?
An abstract class is a class, but not one you can creat objects from directly.
Its purpose is to define how other classes should look like.
```

In [ ]:
```
#21. Can you create an object of an abstract class?
No, we can not creat an object of an abstract class type.
```

In [ ]:
```
#22. Differentiate between data abstraction and encapsulation
Abstraction:
    1)It is a feature of OOPs that hides the unnecessary details but shows the essential information.
    2) It focuses on the external lookout.
Encapsulation:
    1)It is a feature of OOPs that is used to restrict the access of variables and methods
       from outside the class.
       Protecting the data from modification or to prevent accidental damages.
    2) It focuses on internal working.
```

In [ ]:
```
#23. What is polymorphism?
Polymorphism is a very important concept in programming.
It refers to the use of a single type entity (method or object)to represent
different types in different scenarios.

The best example of polymorphism is human behavior. One person can have different behavior.
For example, a person acts as an employee in the office, a customer in the shopping mall,
a passenger in bus/train, a student in school, and a son at home.
```

In [ ]:
```
#24. What is the overloading method?
overriding:
    1)When the method signature (name and parameters) are same in superclass and child class,
      it's called overriding.
    2) When there are same method name of both parent and child class then
       method of child class will always be exicuted.

overloading:
    1) When two or more methods in the same class have the same name but different parameters,
       it's called overloading.
```

In [ ]:
```
#25. What are the limitations of OOPs?
limitation:
    1)Size is larger than other programs
    2)It required a lot of effort to create
    3)It is slower than other programs
    4)It takes time to get used to it
Advantages:
    1)We can reuse the code multiple times using class
    2)It is easy to maintain and modify
    3)It maintains the security of data
    4)Low-cost development
```

In [ ]:
```

```