Concreate class vs Abstract Class vs Inteface:

1. If we dont know anything about implementation just we have requirement specification then we should go for interface.

2. If we are talking about implementation but not completely then we should go for abstract class.(partially implemented class)

3. If we are talking about implementation completely and ready to provide service then we should go for concrete class.

```python
from abc import *
class CollegeAutomation(ABC):
    @abstractmethod
    def m1(self): pass
    @abstractmethod
    def m2(self): pass
    @abstractmethod
    def m3(self): pass
class AbsCls(CollegeAutomation):
    def m1(self):
        print('m1 method implementation')
    def m2(self):
        print('m2 method implementation')

class ConcreteCls(AbsCls):
    def m3(self):
        print('m3 method implemnentation')

c=ConcreteCls()
c.m1()
c.m2()
c.m3()
```

Public, Protected and Private Attributes:

By default every attribute is public. We can access from anywhere either within the class or from outside of the class.

Eg:
name='durga'

Protected attributes can be accessed within the class anywhere but from outside of the class only in child classes. We can specify an attribute as protected by prefexing with _ symbol.

syntax:
_variablename=value