



100%



Title



Arial



26



B

I

U

A



Editing



Python Inheritance Practicals

Note: a) Please put a print("") in each method & init constructor. b) [a few examples](#)

- 1) Create a class called Fruit. Create a constructor and initialize an instance variable name with value 'phalam'. Add a print to indicate as well. Add an instance method called taste() which should print the <name fruit being tasted>.
- 2) Create a class named Apple which subclasses Fruit with an empty body (pass). Now create an object of Apple and invoke taste(). Check if the constructor got executed & if the Fruit's taste() got fired. Do you understand why this happened?
- 3) Now add a constructor in Apple with just a print statement. Now create the object of Apple and double check which constructor got fired. Which one did not? Why so? Now print apple.name and check if the name gets printed. Do you understand why not? Now add the invocation to super().__init__(self) to chain the parent constructor. Now check if the taste() method works properly & whether you can access the name instance variable.
- 4) Add a method called juicify() to Apple and have a print output.
- 5) Create a WashingtonApple that subclasses Apple. Double check if you can invoke all methods of Fruit & Apple on this child object. Now override (same header) taste() in Apple & WashingtonApple and check which body is getting executed when you invoke taste() on Fruit object, Apple object & WashingtonApple object.
- 6) Create a Mango subclass to the Fruit class and override taste() method. What BP should one follow when designing a class? Which dunder should be implemented and why? Create a class called Mangapple which inherits from both Mango & Apple (in that order). Now print Mangapple.mro() or Mangapple.__mro__. Verify the MRO for Mangapple. Print the same for WashingtonApple as well. Now invoke taste() on a Mangapple object. Verify

