|  |
| --- |
| # constructor overloading is not supported # constructor will call which has last precedence   class Eg1:  def \_\_init\_\_(self):  print("Default Constructor")   def \_\_init\_\_(self, a, b, c):  self.a = a  self.b = b  self.c = c  print(self.a, self.b, self.c)    def \_\_init\_\_(self, a, b): # last precedence  self.a = 10  self.b = 20  print(self.a, self.b)  Eg1(10,20)    10 20 |

|  |
| --- |
| # constructor overloading is not supported # default arguments in constructor class Eg2:  def \_\_init\_\_(self, a = 10, b=20):  print(a, b) # 10 20  Eg2() |

|  |
| --- |
| # constructor overloading is not supported # multiple instances class Eg3:  def \_\_init\_\_(self, a = 10, b=20):  print('Values: ', a, b) # 10 20  t1 = Eg3() print(id(t1)) # 313962578944 t2 = Eg3() print(id(t2)) # 688811933120  print(t1 == t2) # False print(t1 is t2) # False |

|  |
| --- |
| # constructor overloading is not supported # arbitrary arguments in constructor class Eg4:  def \_\_init\_\_(self, \*names):  print(names) Eg4("NameOne", "NameTwo", "NameThree")  ('NameOne', 'NameTwo', 'NameThree') |

|  |
| --- |
| # constructor overloading is not supported # arbitrary arguments in constructor class Eg4:  def \_\_init\_\_(self, \*\*colors):  print(colors)  Eg4(c1="Red", c2="Blue", c3 = "Green")  # {'c1': 'Red', 'c2': 'Blue', 'c3': 'Green'} |