



ATHARVA YADAV

ROLL NO : 127

BATCH : S23

INHERITANCE

```

49 class SuperClass:
    tabnine: test | explain | document | ask
50     def super_method(self):
51         print("Super Class method called")
52
53
54 class DerivedClass1(SuperClass):
    tabnine: test | explain | document | ask
55     def derived1_method(self):
56         print("Derived class 1 method called")
57
58 class DerivedClass2(DerivedClass1):
    tabnine: test | explain | document | ask
59     def derived2_method(self):
60         print("Derived class 2 method called")
61
62 d2 = DerivedClass2()
63 d2.super_method()
64 d2.derived1_method()
65 d2.derived2_method()
66
67 class shapes:
    tabnine: test | explain | document | ask
68     def __init__(self, no_sides):
69         self.n = no_sides
70         self.sides = [0 for i in range(no_sides)]
71
    tabnine: test | explain | document | ask
72     def takeSides(self):
73         self.sides = [float(input("Enter side "+str(i+1)+" : ")) for i in range(self.n)]
74
    tabnine: test | explain | document | ask
75     def disSides(self):
76         for i in range(self.n):
77             print("Side",i+1,"is",self.sides[i])
78
79 class rec(shapes):
    tabnine: test | explain | document | ask
80     def __init__(self):
81         shapes.__init__(self,2) # Changed number of sides
82
    tabnine: test | explain | document | ask
83     def findArea(self):
84         a, b = self.sides
85         area = a * b # Changed area calculation
86         print('The area of the rectangle is', area)
87
88 t = rec()
89 t.takeSides()
90
91 mytuple = ("apple", "orange", "grapes") # Changed fruits
92 myit = iter(mytuple)
93 print(next(myit))
94 print(next(myit))
95 print(next(myit))
96
97 mystr = "orange" # Changed fruit
98 myit = iter(mystr)
99 print(next(myit))
100 print(next(myit))
101 print(next(myit))
102 print(next(myit))
103 print(next(myit))
104

```

```

105 mytuple = ("apple", "orange", "grapes") # Changed fruits
106 for x in mytuple:
107     print(x)
108
109 mystr = "orange" # Changed fruit
110 for x in mystr:
111     print(x)
112
113 class MyNumbers:
114     tabnine: test | explain | document | ask
115     def __iter__(self):
116         self.a = 2 # Changed starting number
117         return self
118
119     tabnine: test | explain | document | ask
120     def __next__(self):
121         if self.a <= 22: # Changed upper limit
122             x = self.a
123             self.a += 2 # Changed increment value
124             return x
125         else:
126             raise StopIteration
127
128 myclass = MyNumbers()
129 myiter = iter(myclass)
130 for x in myiter:
131     print(x)
132
133 def myfunc():
134     x = 500 # Changed value
135     print(x)
136
137 myfunc()
138
139 def myfunc():
140     x = 500 # Changed value
141     def myinnerfunc():
142         print(x)
143     myinnerfunc()
144
145 myfunc()
146
147 x = 500 # Changed value
148 def myfunc():
149     print(x)
150
151 myfunc()
152 print(x)
153
154 x = 500 # Changed value
155 def myfunc():
156     x = 400 # Changed local value
157     print(x)
158
159 myfunc()
160 print(x)
161
162 tabnine: test | explain | document | ask
163 def myfunc():
164     global x
165     x = 500 # Changed value
166
167 myfunc()
168 print(x)

```

Aditya Sharma

Welcome Aditya Sharma to the class of 2005

Enter side 1 : [user input]

Enter side 2 : [user input]

Enter side 3 : [user input]

Side 1 is [side 1 value]

Side 2 is [side 2 value]

Side 3 is [side 3 value]

The area of the triangle is [calculated area value]

Super Class method called

Derived class 1 method called

Derived class 2 method called

Enter side 1 : [user input]

Enter side 2 : [user input]

Side 1 is [side 1 value]

Side 2 is [side 2 value]