**Name : jadav savan**

**Roll No : 31**

**PRN : 2017095900001955**

**Sub : Python Programming**

**Sem : 7th**

**Branch : Computer Engineering**

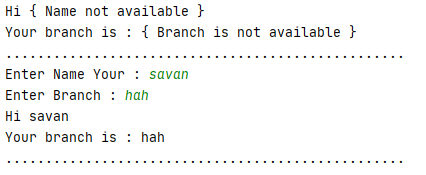
**Practical 11 :**

**(a.) Write a Python Program to demonstrate Working Of classes and objects**

**Solution:**

class Student():  
 def \_\_init\_\_(self, name = **"{ Name not available }"**, branch = **"{ Branch is not available }"**):  
 self.name = name  
 self.branch = branch  
 def display(self):  
 print(**f"Hi** {self.name}**"**)  
 print(**f"Your branch is :** {self.branch}**"**)  
s1 = Student()  
s1.display()  
print(**".................................................."**)  
name = input(**"Enter Name Your : "**)  
branch = input(**"Enter Branch : "**)  
s2 = Student(name, branch)  
s2.display()  
print(**".................................................."**)

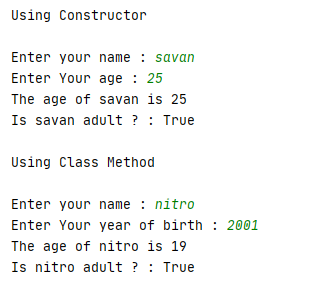
**Output :**



**(b.) Write a Python Program to demonstrate class method and static method**

import datetime  
class Person:  
 def \_\_init\_\_(self, name, age):  
 self.name = name  
 self.age = age  
 @classmethod  
 def fromBirthYear(cls, name, year):  
 return cls(name, datetime.date.today().year - year)  
 @staticmethod  
 def isAdult(age):  
 return age>18  
print(**"**\n**Using Constructor**\n**"**)  
name = input(**"Enter your name : "**)  
age = int(input(**"Enter Your age : "**))  
p1 = Person(name, age)  
print(**f"The age of** {p1.name} **is** {p1.age}**"**)  
  
  
print(**f"Is** {p1.name} **adult ? :** {Person.isAdult(p1.age)}**"**)  
print(**"**\n**Using Class Method**\n**"**)  
name = input(**"Enter your name : "**)  
age = int(input(**"Enter Your year of birth : "**))  
p2 = Person.fromBirthYear(name, age)  
print(**f"The age of** {p2.name} **is** {p2.age}**"**)  
  
  
print(**f"Is** {p2.name} **adult ? :** {Person.isAdult(p2.age)}**"**)

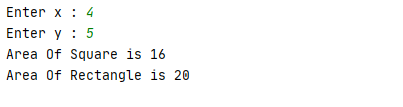
**Output :**

****

**(c.) Write a Python Program To Demonstrate Constructors**

class Square:  
 def \_\_init\_\_(self, x = 0):  
 self.x = x  
 def area(self):  
 print(**f"Area Of Square is** {self.x\*self.x}**"**)  
  
class Rectangle(Square):  
 def \_\_init\_\_(self, x = 0, y = 0):  
 super().\_\_init\_\_(x)  
 self.y = y  
 def area(self):  
 super().area()  
 print(**f"Area Of Rectangle is** {self.x \* self.y}**"**)  
  
x = int(input(**"Enter x : "**))  
y = int(input(**"Enter y : "**))  
r = Rectangle(x,y)  
r.area()

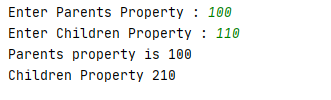
**Output :**



**(d.) Write a python program to demonstrate inheritence**

class Parent:  
 def \_\_init\_\_(self, p = 0):  
 self.property = p  
 def display(self):  
 print(**f"Parents property is** {self.property}**"**)  
class Children(Parent):  
 def \_\_init\_\_(self, p = 0, c = 0):  
 super().\_\_init\_\_(p)  
 self.property1 = c  
 def display(self):  
 super().display()  
 print(**f"Children Property** {self.property1 + self.property}**"**)  
p = int(input(**"Enter Parents Property : "**))  
c = int(input(**"Enter Children Property : "**))  
c = Children(p,c)  
c.display()

**Output :**

****