assesment5

June 11, 2023

```
[ ]: #Q1
```

```
[ ]: 'class is a blueprint of object as it can have nth number of object inside the␣
      ↪given class it as a diff behaviours that object will have in class'
     'object it help to solve the real world promblem created under the class where␣
      ↪it can perform multiple task and  operation '
```

```
[26]: class students_details () :
          def __init__(self,name , email_id ,student_id):
              self.name = name
              self.email_id = email_id
              self.student_id = student_id
          def return_students_detail(self):
              return self.name,self.email_id,self.student_id
```

```
[27]: student = students_details('hcirag','xhura@gmail.com',4335)
```

```
[28]: student.name
```

```
[28]: 'hcirag'
```

```
[29]: student . email_id
```

```
[29]: 'xhura@gmail.com'
```

```
[30]: student . student_id
```

```
[30]: 4335
```

```
[33]: details = student . return_students_detail ()
```

```
[34]: print(details)
```

```
('hcirag', 'xhura@gmail.com', 4335)
```

```
[ ]: #Q2
```

[ ]:

```
. 'polymorphism'
. 'encapsulation'
. 'inheritance'
. 'abstract'
```

[ ]:

```
# Q3
```

[ ]:

```
'__init__ function is used input for the decscribed variable for onces need not␣
 ↪to give a input every time'
```

[35]:

```
class students_details () :
    def __init__(self,name , email_id ,student_id):
        self.name = name
        self.email_id = email_id
        self.student_id = student_id
    def return_students_detail(self):
        return self.name,self.email_id,self.student_id
```

[36]:

```
tudent = students_details('hcirag','xhura@gmail.com',4335)
```

[37]:

```
details = student . return_students_detail ()
```

[38]:

```
print(details)
```

```
('hcirag', 'xhura@gmail.com', 4335)
```

[ ]:

```
#Q4
```

[ ]:

```
' self is used as a pointer to  point a particular  instance of class '
```

[ ]:

```
#Q5
```

[ ]:

```
'inheritance is a fundamental concept of oops . it is defined as the main␣
 ↪parnet class can access the sub class as it is inherited'
```

[89]:

```
' single  inheritance '
```

[89]:

```
' single  inheritance '
```

[90]:

```
class Vehicle:
    def __init__(self, name):
        self.name = name

    def drive(self):
        print(f"{self.name} is being driven.")

class Car (Vehicle):
    def __init__(self, name, color):
        super().__init__(name)
        self.color = color

    def honk(self):
        print(f"{self.name} is honking.")
```

[103]:

```
class Vehicle:
    def __init__(self, name):
        self.name = name

    def drive(self):
        print(f"{self.name} is being driven.")

class Car(Vehicle):
    def __init__(self, name, color):
        super().__init__(name)
        self.color = color

    def honk(self):
        print( "peeeeee ")
car = Car("Toyota fortuner", "Red")
```

[104]:

```
car . name
```

[104]: 'Toyota fortuner'

[105]:

```
car .drive()
```

```
Toyota fortuner is being driven.
```

[106]:

```
car .honk()
```

```
peeeeee
```

[ ]:

```
'multilevel inheritance'
```

[1]:

```
class class1():
    def test_class1(self):
        return 'I am getting attract towoards coding'
```

[2]:

```
class class2(class1):
    def test_class2(self):
        return' oh!! my god help him to get sucess in his life'
```

[3]:

```
class class3(class2):
    def test_class3(self):
        return 'thanks man'
```

```
[4]: obj_class3 = class3()
```

```
[6]: obj_class3.test_class1()
```

```
[6]: 'I am getting attract towoards coding'
```

```
[7]: obj_class3.test_class2()
```

```
[7]: ' oh!! my god help him to get sucess in his life'
```

```
[8]: obj_class3.test_class3()
```

```
[8]: 'thanks man'
```

```
[ ]: 'multiple inhertance'
```

```
[41]: class student :
          def __init__(self , name , email_id) :
                  self.name = name
                  self.email_id = email_id
          def return_student(self):
              return  self.name , self.email_id
      class b :
          def __init__(self ,number):
              self.number = number
          def get_number(self):
              return self.number
      class c(student,b):
          def __init__(self,name ,email_id,number ,student_id):
              student.__init__(self,name,email_id)
              b.__init__(self,number)
              self.student_id = student_id
          def get_student_id(self):
              return self.student_id
```

```
[42]: student21 = c('chirag','chiragn3456@gmail.com',24995803,88403)
```

```
[43]: student21.name
```

```
[43]: 'chirag'
```

[47]:

```
student21.number
```

[47]: 24995803

[48]:

```
student21.get_number()
```

[48]: 24995803

[50]:

```
student21.get_student_id()
```

[50]: 88403

[ ]:

```

```