NAME : M. SAMEED KHAN
ID : 6843
Semester : 8th
Section : B
Submitted to : M. Ayub Khan

Please ANSWER Briefly

Q1 What is class and role of object in class, explain in detail with the help of a suitable program.

ANSWER:

Classes and objects are basic concepts of object oriented programming which revolve around the real life entities. A class is a user defined blueprint or prototype from which objects are created. it represents the set of properties or methods that are common to all object of one type.

* Class: A class has defined as a templete / blueprint. that describe the behaviour / state that the object of its type support.
* Objects: object have states and behaviours.
* EXAMPLE: A dog has states - color name, breed as well as behaviours - wagging the tail.

barking, eating. An object is an instant of a class

* Defining a class in python.

Like functions definitions begins with the def keyword in python. class keyword.

The first string inside the class is called docstring and has a brief description about the class.

As soon as we define a class, a new class object is created with the same name. The class object allows us to access the different attributes as well as to instantiate new objects of that class.

* INPUT:

```
class Person:
    "This is a person class"
    age = 10
    def greet(self):
        print('Hello')
# Output: 10
print(person.age)

# Output: <function Person.greet>
print(person.greet)
# Output: 'This is my second class'
print(person.__doc__)
```

Output:-
10
<function Person.greet at 0x7fc7806
This is a person class

Creating an object in Python:-
This means to say, Since Person.greet is a function object (attribute of class), Person.greet will be a method object.

INPUT:
```
class Person
    "This is a Person"
    age = 10
    def greet(self):
        print('Hello')
# Create a new object of Person class
harry = Person()
# Output: <function Person.greet>
Print(Person.greet)
# Out put: <bound method Person.greet
    of <__main__.Person object>>
Print(harry.greet)
# Calling object's greet() method
# Output: Hello
harry.greet()
```

Output:-
<function Person.greet at 0x7fd7285e4c160>
<bound method Person.greet of <__main__.
Person object at 0x7fd7285ef1307>
Hello

* EXAMPLE
Creating class and object in python

```
class Parrot
    # class attribute
    species = "bird"
    # instance attribute
    def __init__(self, name, age):
        self.name = name
        self.age = age
# Instantiate the parrot class
blu = Parrot("Blu", 10)
woo = Parrot("woo", 15)
# access the class attributes
print("Blu is a {}".format(blu.__class__.species))
Print("woo is also a {}".format(woo.__class__.species))
# access the instance attributes
Print("{} is {} years old".format(blu.name, blu.age))
Print("{} is {} years old".format(woo.name, woo.age))
```

Output
Blu is a bird
woo is also a bird
Blu is 10 years old
woo is 15 years old

Q2 Write a program about table printing which takes input from the user on the basis of OOP and explain in detail.

ANSWER:

Encapsulation
Abstraction
Inheritence
Polymorphism

A. Encapsulation:

The different object about inside of one program will try to communicate with each other automatically. If a programmer wants to stop objects from interacting with each other, they need to be encapsulate in individual classes through the process of encapsulation classes. Cannot change or interact with the specific variables and functions of an object.

Just like a pill form protective barrier about the information that seperate it form the rest of the code. Programmers can replicate this object throughout different parts of the program or other program.

4) Abstraction:-

Abstraction is like an extension of encapsulation because it hides extension properties and methods from the outside code to make the interface of the object simpler. Programmers use abstraction for several beneficial reasons- overall abstraction helps isolate the impact of changes made to the code so that if something goes wrong the change will only affect the variable shown and not the out side code.

5) Inheritance:-

Programmers can extend the functionality of the code existing classes & eliminate repetative code. For instances, element of HTML code that include a text box, select field and checkbox, have certain properties in common with specific method - instead of redefining the properties and methods for each typing of HTML element you can define them once in a generic object. Naming that object something like "HTML elements" will cause either objects to inherit it properties and methods so you reduce unnecessary code.

↳ Polymorphism:

This technique means "many forms are shapes"

allows programmer to render multiple HTML elements depending on the type of object. This concept allow programmers to redefine the way something works by changing how it is done or by changing the parts in which it is done terms of polymorphism are called overriding and overloading

↳ Program in python:

```
# table form (1 to 10) in python

# Taking input from user
table = int (input ("table of"))
# Programs runs from 1 to 10
for i in range (1, 11)
    Print table, "*", i, "=", table*i)
```

↳ Output:

```
table of 5
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
```

Q3 Write a program about any 2 cars which can calculate the performances of both of them and explain in detail.

ANSWER:

* INPUT.

```
Public class Exercise 12 {
public static void main (String[] args) {
  Scanner in = new Scanner (System.in);

System.out.print ("Input speed (KM/H) of first
                   car :");
int car = in.nextInt();

System.out.print("input speed (KM/H) of second
                  car :");
int car2 = in.nextInt();

System.out.println("performance of two
                    cars is :" +
(car1 + car2)/2);.
  }
  }
```

↳ Output:

Sample output:

Input Speed (KM/H) of first car : 100
Input Speed (KM/H) of second car : 80
Average of five numbers is : 90.