

Python Assignment – 2

Name:- Aman Tiwari

Branch :- CSE Final Year

Subject:- Python

Question 01:- *Write a program to implement Polymorphism.*

Solution:-

```
1  class Color:
2      def apply(self):
3          return "Applying Color"
4
5  class Red(Color):
6      def apply(self):
7          return "Applying Red color"
8
9  class Blue(Color):
10     def apply(self):
11         return "Applying Blue color"
12
13 class Green(Color):
14     def apply(self):
15         return "Applying Green color"
16
17 def apply_color(color):
18     print(color.apply())
19
20 red = Red()
21 blue = Blue()
22 green = Green()
23
24 apply_color(red)
25 apply_color(blue)
26 apply_color(green)
```

Output:-

Applying Red color

Applying Blue color

Applying Green color

Question 02:- *Explain the instance, static and class method with example.*

Solutions:-

1:- Instance Method:- An instance method in Python belongs to an object and operates on the instance variables of that object. It takes the “self” parameter as the first argument, representing the instance itself.

Example:-

```
1 class Person:
2     def __init__(self, name, age):
3         self.name = name
4         self.age = age
5
6     def greet(self):
7         return f"Hello, my name is {self.name} and I am {self.age} years old . "
8
9 person1 = Person("Harsh", 20)
10
11 print(person1.greet())
```

02:- Static Method:- A static method in Python is a method that doesn't depend on the object or the class itself. It is defined using the “@staticmethod” decorator and doesn't take the “self” or “cls” parameter by default.

Example :-

```
1 class MathOperations:
2     @staticmethod
3     def multiply(a, b):
4         return a * b
5
6     @staticmethod
7     def subtract(a, b):
8         return a - b
9
10 result_multiply = MathOperations.multiply(5, 3)
11 print("Multiplication Result:", result_multiply)
12
13 result_subtract = MathOperations.subtract(10, 4)
14 print("Subtraction Result:", result_subtract)
```

03 :- Class Method :- A class method in Python is a method that belongs to the class and operates on the class variables. It takes the “cls” parameter as the first argument, representing the class itself. It is defined using the “@classmethod” decorator.

Example :-

```
1 class DateConverter:
2     Date_Format = "YYYY-MM-DD"
3
4     @classmethod
5     def from_string(cls, date_string):
6         year, month, day = map(int, date_string.split('-'))
7         return cls(year, month, day)
8
9     @classmethod
10    def display_date_format(cls):
11        return f"The date format is: {cls.Date_Format}"
12
13    date_object = DateConverter.from_string("2023-11-25")
14    print(date_object.Date_Format)
15
16    print(DateConverter.display_date_format())
```