PW SKILLS

Lesson Plan

Class Methods and Static Methods

Topics to be covered:

- Introduction to Class methods
- Key points of class methods
- Introduction to Static methods
- Key points of static methods

In Python, classes are used to create objects which have properties (attributes) and behaviors (methods). Methods are functions defined inside a class and can be broadly categorized into two types: Class Methods and Static Methods.

Introduction to Class methods

Class methods are methods that are bound to the class and not the instance of the class. They can access or modify class state that applies across all instances of the class. Class methods are defined using the @classmethod decorator.

Example:

```
class Student:
   # Class variable to keep track of the total number of
students
   total_students = 0
   def __init__(self, name):
      self.name = name
      Student.total students += 1


   @classmethod
   def get_total_students(cls):
      return cls.total_students


# Creating student objects
student1 = Student("Alice")
student2 = Student ("Bob")

# Accessing the class method
print("Total students:", Student.get_total_students())
```

Output:

```
Total students: 2
```

- total_students is a class variable that keeps track of the total number of students.
- get_total_students method is a class method because of the @classmethod decorator. It can access the total_students variable and return the total number of students.

Key points of class methods

- Belongs to the Class: Class methods are associated with the class rather than a specific instance. They operate on class-level data that applies to all instances of the class.
- Access to Class Variables: Class methods can access and modify class variables, which are shared by all instances of the class. This makes them useful for tasks that involve manipulating shared data.
- Use Cases: Class methods are often used to create utility functions that perform operations related to the class but don't depend on specific instance data.

Introduction to Static methods

Static methods are methods that belong to the class and don't access or modify class or instance state. They are defined using the @staticmethod decorator.

Example:

```
class Calculator:
   @staticmethod
   def add(x, y):
      return x + y

   @staticmethod
   def subtract(x, y):
      return x - y


# Using static methods
sum_result = Calculator.add(5, 3)


difference result = Calculator.subtract(8, 3)
print("Sum:", sum_result)
print("Difference:", difference_result)
```

Output:

```
Difference: 5
```

- add and subtract are static methods defined inside the Calculator class.
- Static methods don't have access to class or instance variables; they work with the parameters passed to them.

In conclusion, static methods are used when you don't need access to class or instance state, and class methods are used when you need to access or modify class-level attributes. Comprehending these ideas is crucial to writing Python programs that are well-structured and productive. Recall that in Python, class methods are defined with @classmethod decorators and static methods with @staticmethod decorators.