**Worksheet 3: Objects, Classes, and Python Standard Library**

**Objective**: Introduce Python objects, classes, and common modules from the Python Standard Library.

**Estimated Time**: 30 minutes

1. **Introduction (5 minutes)**:
   * Explain what objects and classes are in Python.
   * Define key concepts such as attributes and methods.
   * Provide an overview of useful modules in the Python Standard Library (e.g., math, random, datetime).
2. **Activity 1: Concepts of Classes, Objects, and Modules (10 minutes)**:
   * **Classes**: A blueprint for creating objects (e.g., a Dog class that describes the properties and behaviors of dogs).
     + Example:

python

Copy code

class Dog:

def \_\_init\_\_(self, name, breed):

self.name = name

self.breed = breed

def bark(self):

return f"{self.name} is barking!"

* + **Objects**: Instances of classes, created using the class blueprint.
    - Example:

python

Copy code

my\_dog = Dog("Buddy", "Golden Retriever")

print(my\_dog.bark()) # Output: Buddy is barking!

* + **Modules**: Files that contain Python code (functions, classes, etc.), allowing you to organize code into reusable components. Python has built-in modules like math and random.
    - Example:

python

Copy code

import math

print(math.sqrt(16)) # Output: 4.0

1. **Activity 2: Using the Python Standard Library (10 minutes)**:
   * Import the random module and generate a random number.
   * Use the datetime module to print the current date and time.
   * Use the math module to find the square root of a number.

python

Copy code

import random

print(random.randint(1, 10)) # Random number between 1 and 10

import datetime

print(datetime.datetime.now()) # Current date and time

import math

print(math.sqrt(25)) # Output: 5.0

1. **Reflection Questions (5 minutes)**:
   * What are the benefits of organizing code into classes and modules?
   * How do modules make Python code easier to manage and reuse?