```
#1. Library Access System
class Member:
    def __init__(self, name, member_id):
        self.name = name
        self.member_id = member_id

class StudentMember(Member):
    def __init__(self, name, member_id):
        super().__init__(name, member_id)
        self.borrowed_books = 0

    def add_book(self):
        self.borrowed_books += 1

    def return_book(self):
        if self.borrowed_books > 0:
            self.borrowed_books -= 1

    def display_status(self):
        print(f"{self.name} (ID: {self.member_id}) has borrowed
{self.borrowed_books} book(s).")

# 2. Drone Fleet Management
class Device:
    def basic_device_info(self):
        print("Basic device functions are active.")

class Flyer:
    def fly(self):
        print("Drone is flying.")

class Drone(Device, Flyer):
    def capture_image(self):
        print("Drone is capturing an image.")

# 3. Online Learning Platform
class User:
    def __init__(self, name, email):
        self.name = name
        self.email = email

    def display_info(self):
        print(f"User: {self.name}, Email: {self.email}")

class Instructor(User):
    def __init__(self, name, email, subject):
        super().__init__(name, email)
        self.subject = subject

    def upload_content(self):
        print(f"Instructor {self.name} uploaded content for
{self.subject}.")

    def display_info(self):
        print(f"Instructor: {self.name}, Email: {self.email}, Subject:
{self.subject}")
```

```
class CourseCreator(Instructor):
    def __init__(self, name, email, subject, module_list):
        super().__init__(name, email, subject)
        self.module_list = module_list

    def create_course(self):
        print(f"Course created with modules: {',
'.join(self.module_list)}")

    def display_info(self):
        print(f"Course Creator: {self.name}, Email: {self.email},
Subject: {self.subject}, Modules: {', '.join(self.module_list)}")

# 4. Smart Home Appliance
class Appliance:
    def status(self):
        print("This is a generic appliance.")

class Fan(Appliance):
    def status(self):
        print("Fan is on at speed 3.")

class Light(Appliance):
    def status(self):
        print("Light is set to warm white.")

class AC(Appliance):
    def status(self):
        print("AC is cooling at 24°C.")

# 5. Geometry Toolkit
import math

class ShapeCalculator:
    def area(self, a=None, b=None):
        if a is not None and b is None:
            return math.pi * a * a
        elif a is not None and b is not None:
            return a * b
        else:
            return "Invalid input. Please provide at least one argument."
```