oop-project

Question One (20 Marks)

a) Construct the following classes with proper attributes and methods:

- **Student** class with attributes: `name`, `age`, and `student_id` (2 Marks).
- **Course** class with attributes: `course_name` and `course_code` (2 Marks).

b) Demonstrate how the `Student` class can enroll in and drop courses using methods `enroll(course_name)` and `drop(course_name)` (5 Marks).

c) Create two student objects and enroll each in at least two courses. Then, drop one course for one student (5 Marks).

d) Write a method `display_student_info()` that displays the student's name, ID, and enrolled courses (3 Marks).

e) Ensure your implementation handles edge cases, such as trying to drop a course not enrolled in (3 Marks).

Question Two (20 Marks)

a) Construct the following:

- **Book** class with attributes: `title`, `author`, and `copies_available` (2 Marks).
- **Library** class with a class variable `total_books` to track the total number of books in the library (2 Marks).

b) Demonstrate methods:

- `add_book(title, author, copies)`
- `borrow_book(title)`
- `return_book(title)` within the `Library` class (5 Marks).

c) Add three books to the library and simulate borrowing and returning a book (5 Marks).

d) Display all the books in the library using a method `display_library_info()` (3 Marks).

e) Ensure your program handles cases where a book being borrowed is not available (3 Marks).

Question Three (20 Marks)

a) Construct the following:

- A base class `MenuItem` with attributes: `name`, `price`, and `available` (2 Marks).
- A derived class `Drink` that inherits from `MenuItem` and includes an additional attribute `size` (2 Marks).

- A derived class `Food` that inherits from `MenuItem` and includes an additional attribute `is_vegetarian` (2 Marks).

b) Override the method `order()` in both `Drink` and `Food` classes to display specific order details (5 Marks).

c) Create a class `Order` to add items (both drinks and food) to an order and remove items from the order (5 Marks).

d) Display the final order details and total price using a method `display_order()` (4 Marks).

Question Four (20 Marks)

Design a Smart Home Automation System, focusing on creating classes for smart devices and controlling them within the home.

a) Construct the following classes:

- **Device** class with attributes: `device_name`, `status`, and `location` (2 Marks).
- **SmartHomeController** class with a class variable `total_devices` to track the total number of devices in the home (2 Marks).

b) Demonstrate the following methods in the `SmartHomeController` class (Ensure your program handles cases where a device is not found when trying to turn it on or off, displaying an appropriate message):

- `add_device(device_name, location)` to add a new device to the system.
- `turn_on_device(device_name)` to turn on a specific device.
- `turn_off_device(device_name)` to turn off a specific device (8 Marks).

c) Add three devices to the system and simulate turning on and turning off a device using the above methods (5 Marks).

d) Create a method `display_all_devices()` in the `SmartHomeController` class to display the details of all devices in the system (3 Marks).
