

#### User Class:

- Create a User class that represents both students and instructors.
- Include properties like username, password, email, and user type (student, instructor).

class User:

```
def __init__(self, username, password, email, user_type):  
    self.username = username  
    self.password = password  
    self.email = email  
    self.user_type = user_type
```

#### Course Class:

- Create a Course class to represent individual courses.
- Include properties like course name, course code, instructor, and a list of enrolled students.

class Course:

```
def __init__(self, course_name, course_code, instructor):  
    self.course_name = course_name  
    self.course_code = course_code  
    self.instructor = instructor  
    self.enrolled_students = []
```

#### Assignment Class:

- Create an Assignment class for handling course assignments.
- Include properties like assignment title, description, due date, and associated course.

class Assignment:

```
def __init__(self, title, description, due_date, course):  
    self.title = title  
    self.description = description  
    self.due_date = due_date  
    self.course = course
```

#### Interface for Notifications:

- Define an interface for sending notifications (e.g., email, in-app messages).
- Implement this interface in various notification classes.

```
from abc import ABC, abstractmethod
```

```
class NotificationInterface(ABC):  
    @abstractmethod  
    def send_notification(self, user, message):  
        pass
```

```
class EmailNotification(NotificationInterface):  
    def send_notification(self, user, message):  
        # Implement email notification logic
```

```
class InAppNotification(NotificationInterface):  
    def send_notification(self, user, message):  
        # Implement in-app notification logic
```

### Dashboard Class:

- Create a Dashboard class that acts as the central hub for users.
- Users can view their courses, assignments, and receive notifications.

```
class Dashboard:  
    def __init__(self, user):  
        self.user = user  
        self.courses = [] # List of enrolled courses  
        self.notifications = [] # List of notifications  
  
    def enroll_in_course(self, course):  
        # Implement course enrollment logic  
  
    def view_assignments(self, course):  
        # Implement assignment viewing logic  
  
    def send_notification(self, message, notification_type):  
        # Send notifications using the specified type (email, in-app, etc.)
```

This is a simplified example, and a real-world system like Brightspace would have many more components and complexities. The above structure provides a foundation for managing users, courses, assignments, and notifications in a modular and object-oriented way. It's important to adapt this structure to the specific requirements and technologies you are using in your development.