

## 5. Design a course registration platform

Objects and Behaviors:

Course:

Information: courseName, courseNumber, teacherName, space, maxStudent

Behavior: putOn(), post(), cancel()

Student:

Information: studentName,

Behavior: login(), search(), register(), wait(), delete()

Platform:

Information:

Behavior: sendEmail(), authorize(),

DesignACourseRegistrationPlatform:

Course course[];

Student peter;

Platform neuHome;

```
postCourse(){
    for (int i = 0; i < course.length; i++){
        Course c = course[i];
        c.putOn(neuHome);
        c.post(courseName, courseNumber, teacherName)
    }
}
```

```
registerCourse(){
    peter.login(neuHome);
    peter.search(courseName, courseNumber, teacherName);
    if (space == TRUE){
        peter.register(c);
        maxStudent = maxStudent + 1;
        if (peter.cancel(c) == TRUE){
            maxStudent = maxStudent - 1;
        }
    }else{
        peter.wait(c);
    }
    if (maxStudent < 10){
        c.cancel();
    }
}
```

```
    }  
}  
  
platform(){  
    if (peter.register(c) == TRUE){  
        neuHome.sendEmail("Congratulations! You've registered"+c);  
        neuHome.authorize();  
    }  
}
```