CONCORDIA UNIVERSITY

DEPARTMENT OF
COMPUTER SCIENCE AND SOFTWARE
ENGINEERING

COMP 6231, Fall 2018

Instructor: R. Jayakumar

ASSIGNMENT 2

Distributed Course Registration System (DCRS)
using Java IDL (CORBA)

By

Name : Deep Patel

Student Id : [redacted]

INDEX

1. Methods: (techniques)

1) **addCourse (courseID, semester):**
2) **removeCourse (courseID, semester):**
3) **listCourseAvailability (semester):**
4) **enrolCourse (studentID, courseID, semester):**
5) **dropCourse (studentID, courseID):**
6) **getClassSchedule (studentID):**

7) swapCourse (studentID, newCourseID, oldCourseID)

When an advisor/student invokes this method then…

1. If newCourseID & oldCourseID are his own departments's subjects…

Then server locally gets semester using oldCourseID using getHash() method. Then It drops oldCourse and using semester it enrols newCourse using dropcourseserver() and enrollservermethod() simultaneously. If in some case if newCourse can't be able to enrol them it re-enrol again oldCourse.

e.g. For COMP student swap between COMP691K and COMP 5912

2. If one subject is his own department's subject & 2nd subject is other department's subject…

Then server locally gets semester using CourseID using getHash() method. Then It drops that Course and using semester it enrols Course by invoking appropriate server using inter-server communication. If in some case if newCourse can't be able to enrol them it re-enrol again oldCourse.

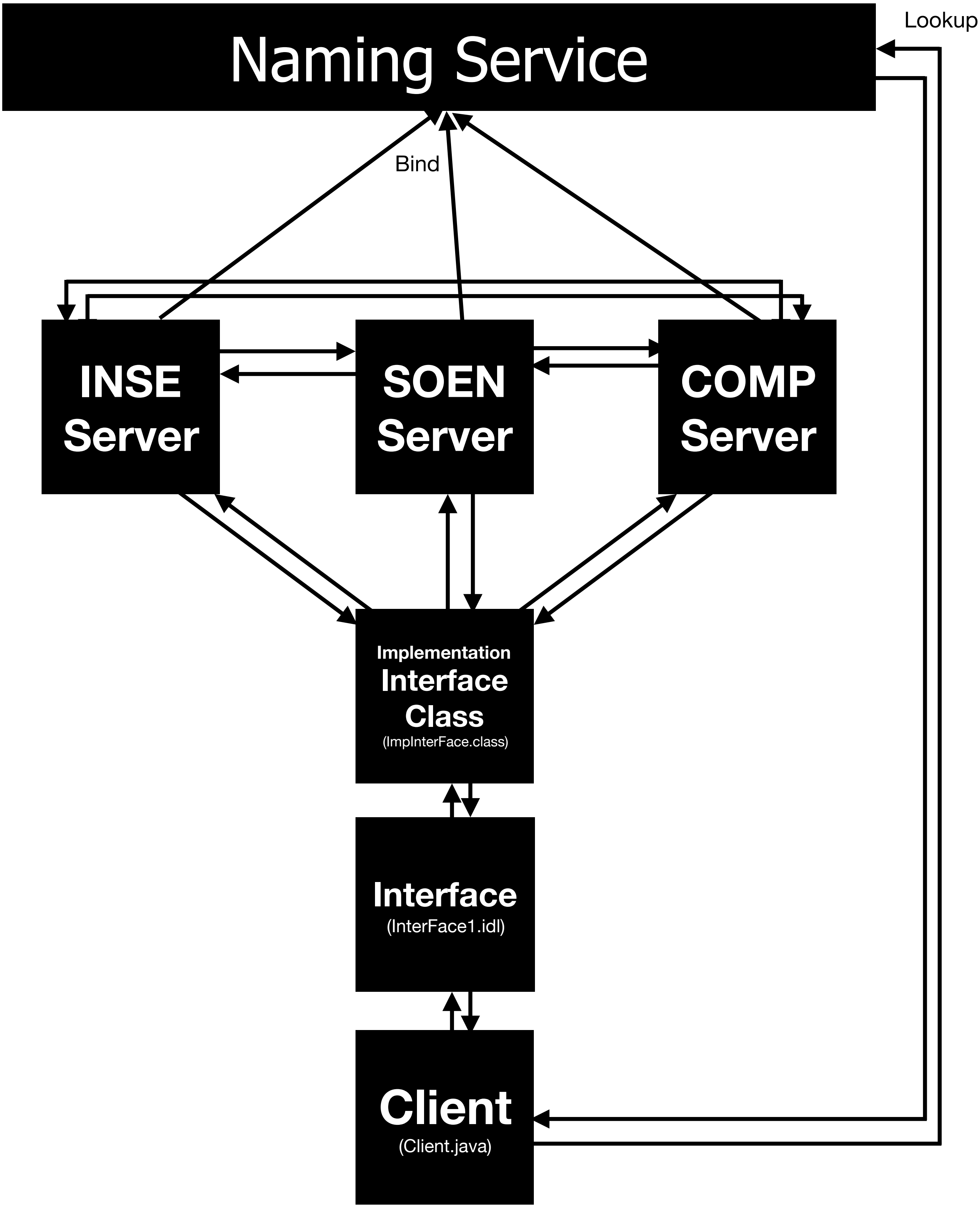
e.g. For COMP student swap between COMP691K and SOEN5912
For COMP student swap between INSE691K and COMP 5912

3. If newCourseID & oldCourseID are his other departments's subjects…

Using inter-server communication server remotely gets semester using oldCourseID using remote getHash() method. Then It drops oldCourse and using semester it enrols newCourse by invoking appropriate server using inter-server communication. If in some case if newCourse can't be able to enrol them it re-enrol again oldCourse.

e.g. For COMP student swap between INSE691K and SOEN5912

2. Architecture of System



Here I used JAVA IDL called CORBA to implement Distributed Course Registration System. In this architecture, servers register itself in naming service this process is called bind. Clients lookup naming registry to find method definition and interface. Then implementation interface invokes the appropriate method in the appropriate server. Using inter-server communication servers exchange information using messages. In the end, Result is ended to the client.

3. Data Structure :

HashMap:

1.

```
HashMap<String, Integer> compfallsubject = new HashMap<String, Integer>();
```

This Hashmap is used to store remain course capacity.

ex. compsummersubject , soenfallsubject , compwintersubject etc.

2.

```
HashMap<String, String> compfallstudent = new HashMap<String, String>();
```

This Hashmap is used to store students information.

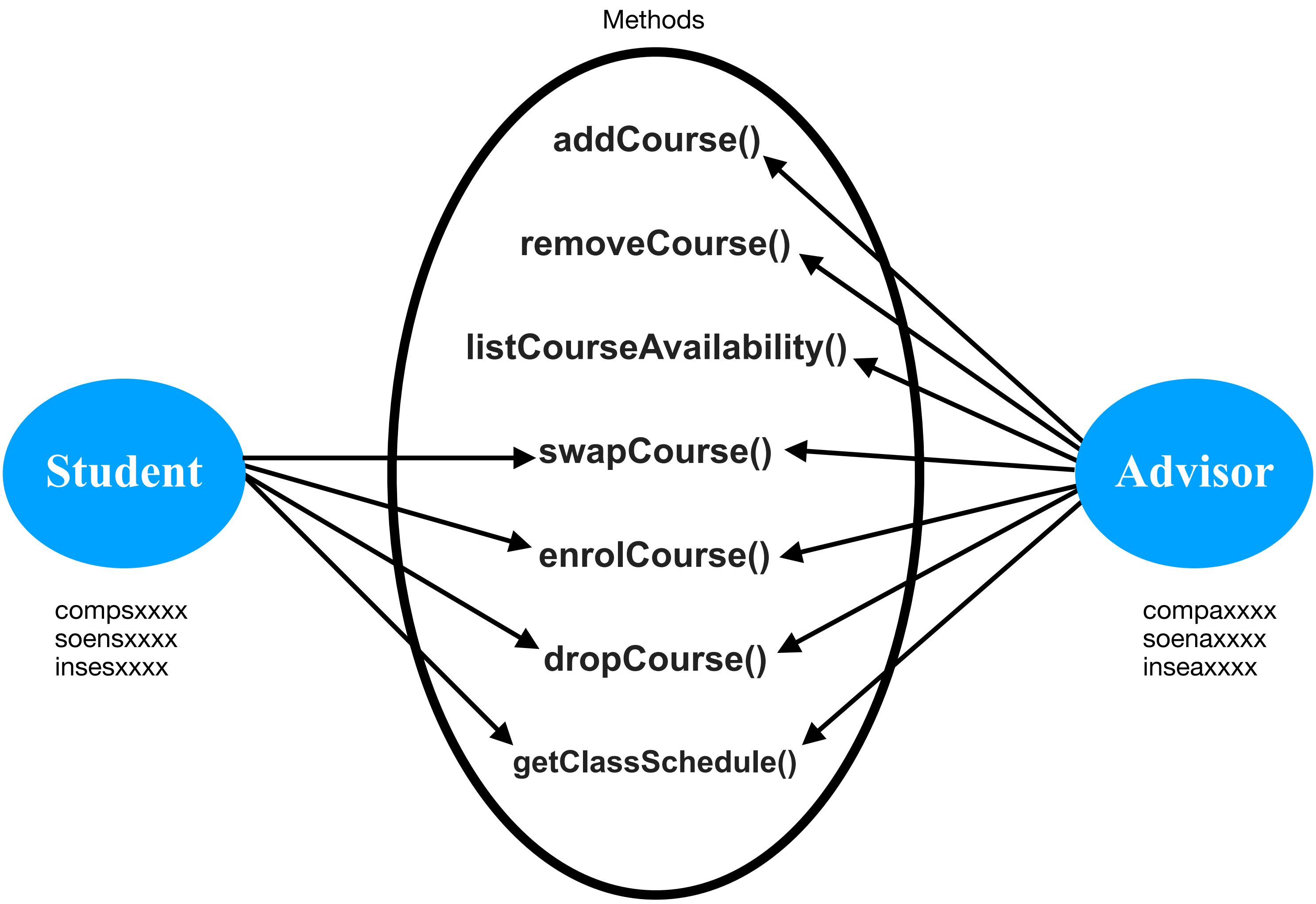
Ex. soenfallstudent , insefallstudent etc.

3.

```
HashMap<String, ArrayList<String>> compadvisor = new HashMap<String,
        ArrayList<String>>();

ArrayList<String> compfallsub = new ArrayList<String>();

ArrayList<String> compwintersub = new ArrayList<String>();

ArrayList<String> compsummersub = new ArrayList<String>();
```

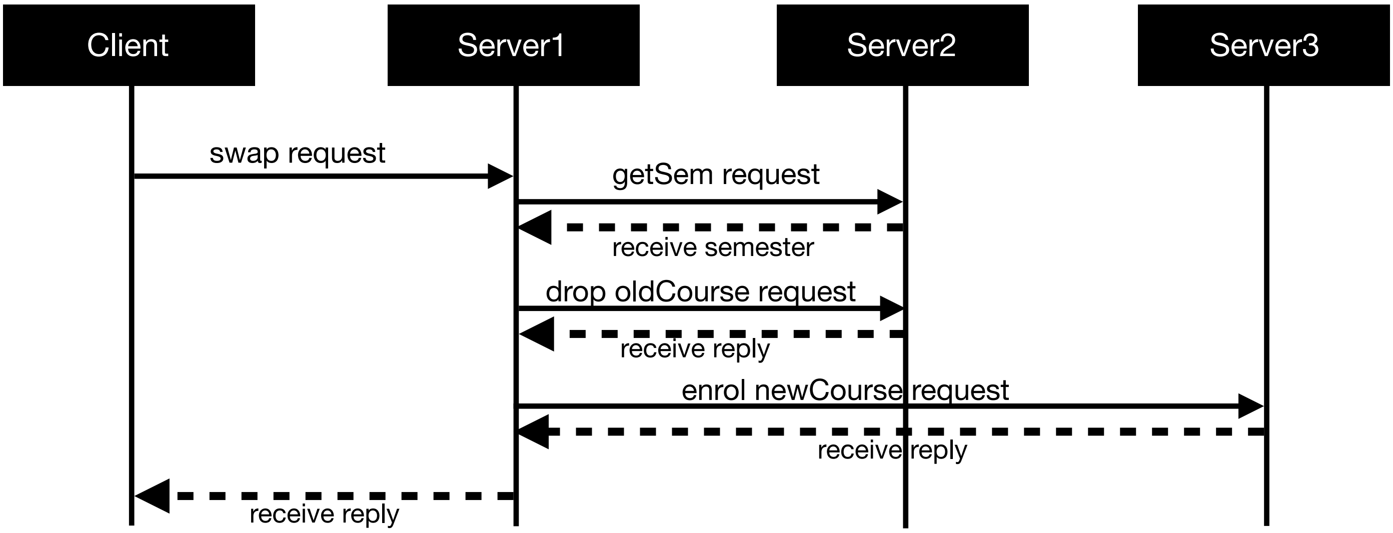
This hashmap is used to store advisor’s information as well as courses added by that advisor in all semesters. Here specific ArrayList stores the courses added by advisors in specific semester.

4. Use Case Diagram:



5. Sequence Diagram:

COMP student wants to swap SOEN course (oldCourse) with INSE course (newCourse)



6. Test cases:

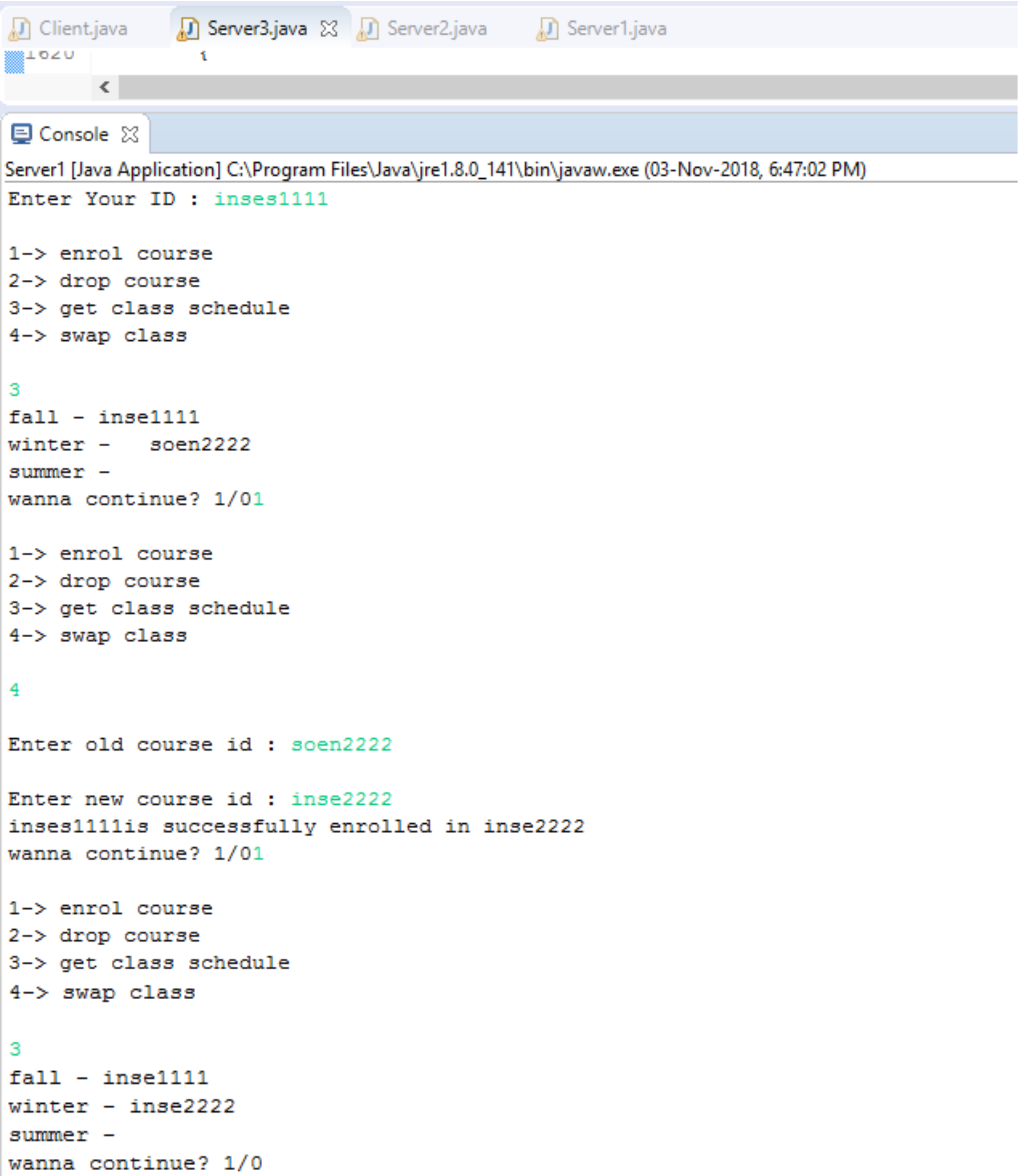
1)

Student id: inses1111

Old course : soen2222

New course : inse2222

Student of INSE department wants to swap course of SOEN department with his own INSE department's course.



Server1 [Java Application] C:\Program Files\Java\jre1.8.0_141\bin\javaw.exe (03-Nov-2018, 6:47:02 PM)

```
Enter Your ID : inses1111

1-> enrol course
2-> drop course
3-> get class schedule
4-> swap class

3
fall - inse1111
winter -   soen2222
summer -
wanna continue? 1/01

1-> enrol course
2-> drop course
3-> get class schedule
4-> swap class

4

Enter old course id : soen2222

Enter new course id : inse2222
inses1111is successfully enrolled in inse2222
wanna continue? 1/01

1-> enrol course
2-> drop course
3-> get class schedule
4-> swap class

3
fall - inse1111
winter - inse2222
summer -
wanna continue? 1/0
```

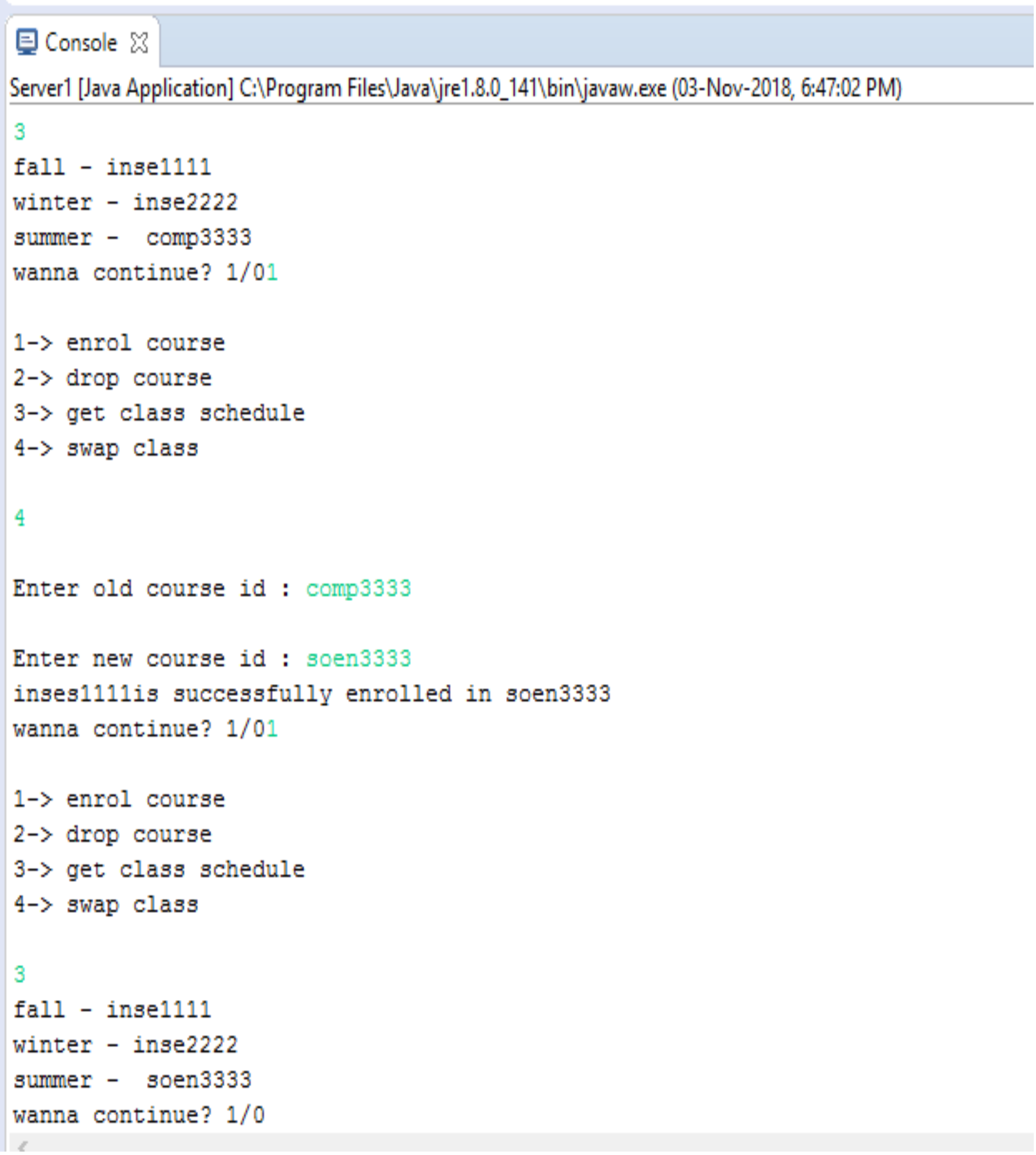
2)

Student id: inses1111

Old course : comp3333

New course : soen3333

Student of INSE department wants to swap course of COMP department with SOEN department's course.



```
Server1 [Java Application] C:\Program Files\Java\jre1.8.0_141\bin\javaw.exe (03-Nov-2018, 6:47:02 PM)
3
fall - inse1111
winter - inse2222
summer -  comp3333
wanna continue? 1/01

1-> enrol course
2-> drop course
3-> get class schedule
4-> swap class

4

Enter old course id : comp3333

Enter new course id : soen3333
inses1111is successfully enrolled in soen3333
wanna continue? 1/01

1-> enrol course
2-> drop course
3-> get class schedule
4-> swap class

3
fall - inse1111
winter - inse2222
summer -  soen3333
wanna continue? 1/0
```

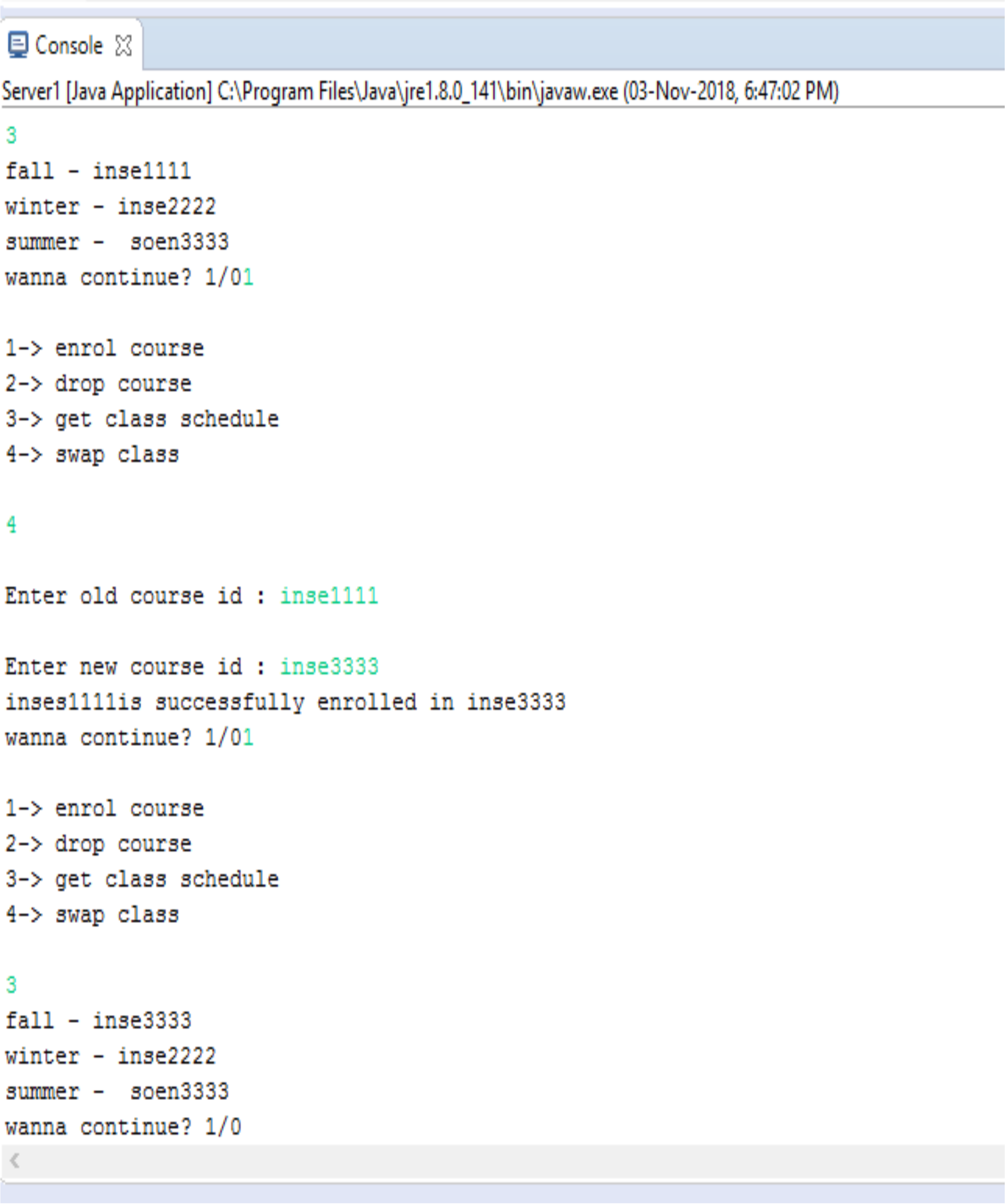
3)

Student id: inses1111

Old course : inse1111

New course : inse3333

Student of INSE department wants to swap 2 courses of his own department.



```
Server1 [Java Application] C:\Program Files\Java\jre1.8.0_141\bin\javaw.exe (03-Nov-2018, 6:47:02 PM)
3
fall - inse1111
winter - inse2222
summer -  soen3333
wanna continue? 1/01

1-> enrol course
2-> drop course
3-> get class schedule
4-> swap class

4

Enter old course id : inse1111

Enter new course id : inse3333
inses1111is successfully enrolled in inse3333
wanna continue? 1/01

1-> enrol course
2-> drop course
3-> get class schedule
4-> swap class

3
fall - inse3333
winter - inse2222
summer -  soen3333
wanna continue? 1/0
```

7. Important/difficult part :

One difficult part of this assignment is inter-server communication in CORBA. I faced trouble many time while sending data from one server to another server. There was a small problem of how to send different information in one single message and how to separate such information like user id, semester etc in the received message on another server. I faced problems in maintaining server log files also. Most important part of the assignment is the swap method, in this operation, remote method needs to invoke many time using inter-server communication. To enrol newCourse we must need a semester, which can be fetched using oldCourseID. It is also important that if in some case if newCourse can’t enrol then the student must not be dropped in oldCourse at end of the operation.