

Main.java

```

1 package set10111.simulation;
2 import java.lang.reflect.Array;
16
17
18 public class Main {
19
20
21     private static ArrayList<String> studentAgents = new ArrayList<>();
22     static ArrayList<String> moduleNames = new ArrayList<String>();
23     private static HashMap<String, Student> studentList = new HashMap<String, Student>();
24 // static HashMap<String, TimeSlot> myPreferences = new HashMap<String, TimeSlot>();
25     static ArrayList<HashMap<String, String>> allPrefs = new
    ArrayList<HashMap<String,String>>();
26     static Random random = new Random();
27
28
29 // Note : Only owner of the time slot to be swapped should tell timetabling agent to swap
    the slots?
30     // what should be included in the Request from the student who want other agent to
    swap the slot when other student confirm that he is the owner of enquired slot in the
    previous call
31     // I mean request before sending final swap action.
32
33
34 // once state updated by timetabling agent should timetabling re-send the timetable to
    concerned students or should student update their timetable locally themselves. It is not
    preferences but a copy of
35     // student's timetable stored locally in student agent class which is sent by the
    timetabling agent when system starts
36
37     // ask should we have preferences separately for each module or just once
38
39
40     //You should pass preferences to student agent here in main function
41     // Also can generate time table for tutorials here in main and then pass it to
    timetabling agent
42
43     public static void main(String[] args) {
44         Profile myProfile = new ProfileImpl();
45         Runtime myRuntime = Runtime.instance();
46 // ArrayList<String> moduleNames = new ArrayList<String>();
47 // HashMap<String, Student> studentList = new HashMap<String, Student>();
48         Random random = new Random();
49         try{
50             ContainerController myContainer = myRuntime.createMainContainer(myProfile);
51             AgentController rma = myContainer.createNewAgent("rma", "jade.tools.rma.rma",
    null);
52             rma.start();
53 //***** Run with testing
54             // change the number when testing different test cases to set to the number of
    agents involved in test case.
55             // It is only used for test cases, in normal scenario it uncommment run without
    test code
56             int numStudents = 2;
57             for (int i = 0; i < numStudents; i++) {
58                 studentAgents.add(""+i);
59                 HashMap<String, String> pref= PreferenceGenerator2();
60
61                 allPrefs.add(pref);
62             }
63
64

```

Main.java

```
65         // for 2 modules change the module assignment to 2 modules in TA agent.
66 //         TestNoSwapAlreadySatisfied();
67 //         Test2Modules2Swap();
68
69         // 2-agents
70 //         Test3Modules3Swap();
71         Test3ModulesButSwap2();
72 //         Test3ModulesButSwap1();
73 //         Test3ModulesNoSwapNeeded();
74
75         // all deal with 3 agents
76 //         _3Modules_3Agents_only_2SwapSlots();
77
78
79
80 //         create specified number of student agents without hardcoding every single
        student agent
81         AgentController studentAgent;
82         for(int i=0; i<numStudents; i++) {
83             int studentIdentifier = i+1;
84
85             Customm myPref = new Customm();
86             myPref.preferences = allPrefs.get(i);
87
88             Customm[] pass = {myPref};
89             String studentName = "Student" + studentIdentifier;
90             studentAgent = myContainer.createNewAgent("Student" + studentIdentifier,
StudentAgent.class.getCanonicalName(), pass);
91             studentAgent.start();
92
93 //             studentAgents.set(i, studentName);
94         }
95
96 //         generateTimetable();
97         Customm nn = new Customm();
98         nn.timetable = studentList;
99         Customm[] c = {nn};
100         System.out.println(c);
101
102
103         AgentController timetablingAgent =
myContainer.createNewAgent("timetablingAgent1", TimeTablingAgent.class.getCanonicalName(),
c);
104         timetablingAgent.start();
105
106
107 //***** Run with testing end
108
109 //*****
110 //         Run without testing
111
112 //
113 //         AgentController timetablingAgent =
myContainer.createNewAgent("timetablingAgent1", TimeTablingAgent.class.getCanonicalName(),
null);
114 //         timetablingAgent.start();
115 //
116 //
117 //         AgentController tickerAgent = myContainer.createNewAgent("ticker",
BuyerSellerTicker.class.getCanonicalName(),
118 //             null);
119 //         tickerAgent.start();
```

Main.java

```

120 //
121 //
122 //     AgentController studentAgent;
123 //     int numStudents = 2;
124 //     for(int i=0; i<numStudents; i++) {
125 //         int studentIdentifier = i+1;
126 //         studentAgent = myContainer.createNewAgent("Student" + studentIdentifier,
127 //             StudentAgentNew.class.getCanonicalName(), null);
128 //         studentAgent.start();
129 //     }
130 //
131 //     //***** Moved to timetabling agent *****
132 //
133 //     // assign random time slots and random tutorial groups to each student for
134 //     each module
135 //     for (String name : moduleNames) {
136 //         for (Student student : studentList) {
137 //             int groupId = random.nextInt(2);           // generate random number
138 //             between 0 and 1
139 //             Date day = new Date(2);
140 //
141 //             int startTime = random.nextInt(9) + 9;
142 //             int endTime = startTime + 1;
143 //
144 //             TimeSlot slot = new TimeSlot(name, groupId, day, startTime, endTime,
145 //                 "AssignedByTA");
146 //
147 //             Module module = new Module(name, slot);
148 //
149 //             // add this slot to the module list of student
150 //             student.moduleList.add(module);
151 //
152 //             System.out.println(student.name);
153 //             System.out.println(" slot : " + slot.getModuleName() + " " +
154 //                 slot.getGroupId() + " " + slot.getDate() + " " + slot.getStartTime()+ " " +
155 //                 slot.getEndTime());
156 //         }
157 //     }
158 //
159 //     }
160 //
161 //     }
162 //
163 //     private static void timetableForTesting() {
164 //         HashMap<String, Student> studentList = new HashMap<>();
165 //
166 //         //create student1
167 //         Student student = new Student();
168 //         String sa = studentAgents.get(0);
169 //         student.name = sa;
170 //         student.moduleList = new HashMap<String,TimeSlot>();
171 //
172 //         String moduleName = "Module1";
173 //
174 //
175 //

```

Main.java

```
176         int startTime = random.nextInt(9) + 9;
177         int endTime = startTime + 1;
178         int day = random.nextInt(5) + 1; // set the day of the tutorial
179         TimeSlot slot = new TimeSlot(); slot.setModuleName(moduleName);
        slot.setGroupId(0); // set to 0 because will be updated when actual timeslot is assigned
180         slot.setDate(day); slot.setStartTime(startTime); slot.setEndTime(endTime);
        slot.setStatus("AssignedByTA");
181
182         student.moduleList.put(moduleName, slot);
183
184         // add to student list
185         studentList.put(sa, student);
186
187
188
189
190         //Student 2
191         student = new Student();
192         sa = studentAgents.get(1);
193         student.name = sa;
194         student.moduleList = new HashMap<String,TimeSlot>();
195
196
197         moduleName = "Module1";
198
199
200         startTime = random.nextInt(9) + 9;
201         endTime = startTime + 1;
202         day = random.nextInt(5) + 1; // set the day of the tutorial
203         slot = new TimeSlot(); slot.setModuleName(moduleName); slot.setGroupId(0); // set
        to 0 because will be updated when actual timeslot is assigned
204         slot.setDate(day); slot.setStartTime(startTime); slot.setEndTime(endTime);
        slot.setStatus("AssignedByTA");
205
206         student.moduleList.put(moduleName, slot);
207
208         // add to student list
209         studentList.put(sa, student);
210     }
211
212     private static Customm generateTimetable() {
213         studentList = new HashMap<String, Student>();
214
215         moduleNames.clear();
216
217         // add new dummy module
218         moduleNames.add("Module1");
219         moduleNames.add("Module2");
220         moduleNames.add("Module3");
221
222         // create students list based upon number of student agents in system
223         for (String sa : studentAgents) {
224             Student student = new Student();
225             student.name = sa;
226             student.moduleList = new HashMap<String,TimeSlot>();
227
228             // add to student list
229             studentList.put(sa, student);
230
231         }
232
233         // Now assign random time slots and random tutorial groups to each student for
```

Main.java

```

234 // each module
235 // for (String moduleName : moduleNames) {
236 //     System.out.println("");
237 //     System.out.println(moduleName);
238 //     System.out.println("-----");
239 //
240 //     int totalStudentG1 = 0;
241 //     int totalStudentG2 = 0;
242
243 // generate 2 time slots for each tutorial
244 // ArrayList<TimeSlot> tutTime = new ArrayList<TimeSlot>();
245 // for (int i = 0; i < 2; i++) {
246 //
247 //     int startTime = random.nextInt(9) + 9;
248 //     int endTime = startTime + 1;
249 //     int day = random.nextInt(5) + 1; // set the day of the tutorial
250 //
251 //     TimeSlot slot = new TimeSlot();
252 //     slot.setModuleName(moduleName);
253 //     slot.setGroupId(0); // set to 0 because will be updated when actual
    timeslot is assigned
254 //     slot.setDate(day);
255 //     slot.setStartTime(startTime);
256 //     slot.setEndTime(endTime);
257 //     slot.setStatus("AssignedByTA");
258 //     tutTime.add(slot); // adding dummy slot to tutorial times arraylist
259 // }
260 //
261 // // assign each student to random timeslot of tutorial
262 // for (String aid : studentAgents) {
263 //     Student student = studentList.get(aid);
264 //
265 //     int groupId = random.nextInt(2); // generate random number between 0 and 1
266 //
267 //     TimeSlot slot = tutTime.get(groupId);
268 //     slot.setGroupId(groupId + 1); // assign the group id to slot. adding one
    because want tp statr
269 //                                     // group ids from 1 instead of 0
270 //
271 //
272 //     // add this slot to the module list of student
273 //     studentList.get(aid).moduleList.put(moduleName, slot);
274 //
275 //     if (groupId == 0) {
276 //         totalStudentG1++;
277 //     } else {
278 //         totalStudentG2++;
279 //     }
280 //
281 //     System.out.println(student.name);
282 //     System.out.println(
283 //         student.name + "    ----    slot : " + slot.getModuleName() + "
    " + slot.getGroupId()
284 //         + " " + slot.getDate() + " " + slot.getStartTime() + "
    " + slot.getEndTime());
285 //     }
286
287
288
289 //     System.out.println("Total Students enrolled on module : " +
    studentAgents.size());
290 //     System.out.println(" student in Group 1 : " + totalStudentG1);

```

Main.java

```

291 //      System.out.println(" student in Group 2 : " + totalStudentG2);
292 //    }
293
294
295    Customm testTimetable = new Customm();
296    testTimetable.timetable = studentList;
297    return testTimetable;
298  }
299
300
301  public static HashMap<String, String> PreferenceGenerator(){
302
303    HashMap<String, String> myPreferences = new HashMap<String, String>();
304    try {
305
306      ArrayList<String> prefCodes = new ArrayList<String>();
307      prefCodes.add("outside work or caring commitments");
308      prefCodes.add("prefer not to attend");
309      prefCodes.add("like to have");
310
311
312      for (int i = 0; i < 5; i++) { // add preferences for 5 days
313
314        int day = i + 1;
315
316        // assign time slots for each day for hours between 9-17
317        for (int j = 9; j < 18; j++) {
318
319          int random = (int) Math.round((0 + 2 * Math.random())); // generate
random number between 0
320
321          int startTime = j;
322          int endTime = j + 1;
323
324          String preferenceCode = prefCodes.get(random); // get random
preference
325
326          TimeSlot slot = new TimeSlot();
327          slot.setModuleName("");
328          slot.setGroupId(9); // should be optional because enot needed for
preferences
329
330          slot.setDate(day);
331          slot.setStartTime(startTime);
332          slot.setEndTime(endTime);
333          slot.setStatus(preferenceCode);
334
335          String slotID = String.valueOf(day).concat(String.valueOf(startTime))
.concat(String.valueOf(endTime));
336
337          myPreferences.put(slotID, preferenceCode);
338        }
339
340      }
341
342    } catch (Exception e) {
343      e.printStackTrace();
344    }
345
346
347    return myPreferences;
348  }
349

```

```

350
351
352     private static void Test1(HashMap<String, Student> studentList, HashMap<String,
String> pref1, HashMap<String, String> pref2) {
384
385
386     public static HashMap<String, String> PreferenceGenerator2(){
427
428
429     private static void Test2() {
430
431         //test 1
432         student1
433         TimeSlot slot=
studentList.get(studentAgents.get(0)).moduleList.get(moduleNames.get(0));
434         String slotId1 =
String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
alueOf(slot.getEndTime()));
435 //
436 //
437 //         //student 2
438 //         TimeSlot slot2=
studentList.get(studentAgents.get(1)).moduleList.get(moduleNames.get(0));
439 //         String slotId2 =
String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
440
441
442 //         // make clash with preferences by setting the preferences to unable to attend
443 //         String s1 = pref1.get(slotId1);
444 //         String s2 = pref2.get(slotId1);
445 //         pref1.replace(slotId1, "outside work or caring commitments");
446 //         String s3 = pref1.get(slotId1);
447 //         String s4 = pref2.get(slotId1);
448
449         // make clash with preferences by setting the preferences to unable to attend
450         // make clash with preferences by setting the preferences to unable to attend
451 //         pref2.get(slotId2).setStatus("outside work or caring commitments");
452 //         pref1.get(slotId2).setStatus("like to have");
453 //         pref2.replace(slotId2, "outside work or caring commitments");
454
455 //         // now set the id of other agent to different tutorial so that they can swap
456 //         slot.setGroupId(1);
457 //         slot2.setGroupId(2);
458
459         studentList = new HashMap<String, Student>();
460         Student student = new Student();
461         student.name = "Student1";
462         student.moduleList = new HashMap<String, TimeSlot>();
463
464         // add to student list
465         studentList.put("Student1", student);
466
467
468         TimeSlot slot = new TimeSlot();
469         slot.setModuleName("Module1");slot.setGroupId(2);slot.setDate(1);slot.setStartTime
(12);slot.setEndTime(13);slot.setStatus("Assigned By Ta");
470         String slotId1 =
String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
alueOf(slot.getEndTime()));
471         allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
472

```

Main.java

```

473     studentList.get("Student1").moduleList.put("Module1", slot);
474
475
476
477
478     Student student2 = new Student();
479     student2.name = "Student2";
480     student2.moduleList = new HashMap<String,TimeSlot>();
481
482     // add to student list
483     studentList.put("Student2", student2);
484
485     TimeSlot slot2 = new TimeSlot();
486     slot2.setModuleName("Module1");slot2.setGroupId(1);slot2.setDate(1);slot2.setStart
Time(9);slot2.setEndTime(10);slot2.setStatus("Assigned By Ta");
487     String slotId2 =
String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
488     allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
489
490     studentList.get("Student2").moduleList.put("Module1", slot2);
491 }
492
493
494 private static void TestNoSwapAlreadySatisfied() {
495
496     studentList = new HashMap<String, Student>();
497
498     //Student 1
499     Student student = new Student();
500     student.name = "Student1";
501     student.moduleList = new HashMap<String,TimeSlot>();
502     // add to student list
503     studentList.put("Student1", student);
504
505     //Student 2
506     Student student2 = new Student();
507     student2.name = "Student2";
508     student2.moduleList = new HashMap<String,TimeSlot>();
509     // add to student list
510     studentList.put("Student2", student2);
511
512
513
514
515
516     //add slot
517     TimeSlot slot = new TimeSlot();
518     slot.setModuleName("Module1");slot.setGroupId(2);slot.setDate(1);slot.setStartTime
(12);slot.setEndTime(13);slot.setStatus("Assigned By Ta");
519     String slotId1 =
String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
alueOf(slot.getEndTime()));
520 //     allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
521
522     studentList.get(student.name).moduleList.put("Module1", slot);
523
524
525
526
527
528     TimeSlot slot2 = new TimeSlot();

```


Main.java

```
529         slot2.setModuleName("Module1");slot2.setGroupId(1);slot2.setDate(1);slot2.setStart
Time(9);slot2.setEndTime(10);slot2.setStatus("Assigned By Ta");
530         String slotId2 =
String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
531 //         allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
532
533         studentList.get(student2.name).moduleList.put("Module1", slot2);
534     }
535
536
537
538 private static void Test2Swap1With3Agents() {
539
540     studentList = new HashMap<String, Student>();
541     Student student = new Student();
542     student.name = "Student1";
543     student.moduleList = new HashMap<String,TimeSlot>();
544
545     // add to student list
546     studentList.put("Student1", student);
547
548
549     TimeSlot slot = new TimeSlot();
550     slot.setModuleName("Module1");slot.setGroupId(2);slot.setDate(1);slot.setStartTime(12)
;slot.setEndTime(13);slot.setStatus("Assigned By Ta");
551     String slotId1 =
String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
alueOf(slot.getEndTime()));
552     allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
553
554     studentList.get("Student1").moduleList.put("Module1", slot);
555
556
557
558
559     Student student2 = new Student();
560     student2.name = "Student2";
561     student2.moduleList = new HashMap<String,TimeSlot>();
562
563     // add to student list
564     studentList.put("Student2", student2);
565
566     TimeSlot slot2 = new TimeSlot();
567     slot2.setModuleName("Module1");slot2.setGroupId(1);slot2.setDate(1);slot2.setStartTime
(9);slot2.setEndTime(10);slot2.setStatus("Assigned By Ta");
568     String slotId2 =
String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
569     allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
570
571     studentList.get("Student2").moduleList.put("Module1", slot2);
572
573
574     Student student3 = new Student();
575     student3.name = "Student3";
576     student3.moduleList = new HashMap<String,TimeSlot>();
577
578     // add to student list
579     studentList.put("Student3", student3);
580
581
```

Main.java

```
582     TimeSlot slot3 = new TimeSlot();
583     slot3.setModuleName("Module1");slot3.setGroupId(2);slot3.setDate(3);slot3.setStartTime
    (16);slot3.setEndTime(17);slot3.setStatus("Assigned By Ta");
584     String slotId3 =
        String.valueOf(slot3.getDate()).concat(String.valueOf(slot3.getStartTime())).concat(String
        .valueOf(slot3.getEndTime()));
585     allPrefs.get(2).replace(slotId3, "outside work or caring commitments");
586
587     studentList.get("Student3").moduleList.put("Module1", slot3);
588 }
589
590
591
592 private static void Test2Modules2Swap() {
593
594
595     studentList = new HashMap<String, Student>();
596     Student student = new Student();
597     student.name = "Student1";
598     student.moduleList = new HashMap<String,TimeSlot>();
599
600     // add to student list
601     studentList.put("Student1", student);
602
603
604     TimeSlot slot = new TimeSlot();
605     slot.setModuleName("Module1");slot.setGroupId(2);slot.setDate(1);slot.setStartTime(12)
    ;slot.setEndTime(13);slot.setStatus("Assigned By Ta");
606     String slotId1 =
        String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
        alueOf(slot.getEndTime()));
607     allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
608
609     studentList.get("Student1").moduleList.put("Module1", slot);
610
611
612     slot = new TimeSlot();
613     slot.setModuleName("Module2");slot.setGroupId(2);slot.setDate(5);slot.setStartTime(12)
    ;slot.setEndTime(13);slot.setStatus("Assigned By Ta");
614     slotId1 =
        String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
        alueOf(slot.getEndTime()));
615     allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
616
617     studentList.get("Student1").moduleList.put("Module2", slot);
618
619
620
621
622     Student student2 = new Student();
623     student2.name = "Student2";
624     student2.moduleList = new HashMap<String,TimeSlot>();
625
626     // add to student list
627     studentList.put("Student2", student2);
628
629     TimeSlot slot2 = new TimeSlot();
630     slot2.setModuleName("Module1");slot2.setGroupId(1);slot2.setDate(1);slot2.setStartTime
    (9);slot2.setEndTime(10);slot2.setStatus("Assigned By Ta");
631     String slotId2 =
        String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
        .valueOf(slot2.getEndTime()));
```

```

632     allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
633
634     studentList.get("Student2").moduleList.put("Module1", slot2);
635
636     slot2 = new TimeSlot();
637     slot2.setModuleName("Module2");slot2.setGroupId(1);slot2.setDate(4);slot2.setStartTime
(9);slot2.setEndTime(10);slot2.setStatus("Assigned By Ta");
638     slotId2 =
String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
639     allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
640
641     studentList.get("Student2").moduleList.put("Module2", slot2);
642 }
643
644 private static void Test3Modules3Swap() {
645
646
647     studentList = new HashMap<String, Student>();
648     Student student = new Student();
649     student.name = "Student1";
650     student.moduleList = new HashMap<String,TimeSlot>();
651
652     // add to student list
653     studentList.put("Student1", student);
654
655
656     TimeSlot slot = new TimeSlot();
657     slot.setModuleName("Module1");slot.setGroupId(2);slot.setDate(1);slot.setStartTime(12)
;slot.setEndTime(13);slot.setStatus("Assigned By Ta");
658     String slotId1 =
String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
alueOf(slot.getEndTime()));
659     allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
660
661     studentList.get("Student1").moduleList.put("Module1", slot);
662
663
664     slot = new TimeSlot();
665     slot.setModuleName("Module2");slot.setGroupId(2);slot.setDate(5);slot.setStartTime(12)
;slot.setEndTime(13);slot.setStatus("Assigned By Ta");
666     slotId1 =
String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
alueOf(slot.getEndTime()));
667     allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
668
669     studentList.get("Student1").moduleList.put("Module2", slot);
670
671     slot = new TimeSlot();
672     slot.setModuleName("Module3");slot.setGroupId(2);slot.setDate(2);slot.setStartTime(12)
;slot.setEndTime(13);slot.setStatus("Assigned By Ta");
673     slotId1 =
String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
alueOf(slot.getEndTime()));
674     allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
675
676     studentList.get("Student1").moduleList.put("Module3", slot);
677
678
679
680
681     Student student2 = new Student();

```

```

682     student2.name = "Student2";
683     student2.moduleList = new HashMap<String, TimeSlot>();
684
685     // add to student list
686     studentList.put("Student2", student2);
687
688     TimeSlot slot2 = new TimeSlot();
689     slot2.setModuleName("Module1"); slot2.setGroupId(1); slot2.setDate(1); slot2.setStartTime
(9); slot2.setEndTime(10); slot2.setStatus("Assigned By Ta");
690     String slotId2 =
String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
691     allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
692
693     studentList.get("Student2").moduleList.put("Module1", slot2);
694
695     slot2 = new TimeSlot();
696     slot2.setModuleName("Module2"); slot2.setGroupId(1); slot2.setDate(4); slot2.setStartTime
(9); slot2.setEndTime(10); slot2.setStatus("Assigned By Ta");
697     slotId2 =
String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
698     allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
699
700     studentList.get("Student2").moduleList.put("Module2", slot2);
701
702     slot2 = new TimeSlot();
703     slot2.setModuleName("Module3"); slot2.setGroupId(1); slot2.setDate(3); slot2.setStartTime
(15); slot2.setEndTime(16); slot2.setStatus("Assigned By Ta");
704     slotId2 =
String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
705     allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
706
707     studentList.get("Student2").moduleList.put("Module3", slot2);
708 }
709
710
711 private static void Test3ModulesButSwap2() {
712
713
714     studentList = new HashMap<String, Student>();
715     Student student = new Student();
716     student.name = "Student1";
717     student.moduleList = new HashMap<String, TimeSlot>();
718
719     // add to student list
720     studentList.put("Student1", student);
721
722
723     TimeSlot slot = new TimeSlot();
724     slot.setModuleName("Module1"); slot.setGroupId(2); slot.setDate(1); slot.setStartTime(12)
; slot.setEndTime(13); slot.setStatus("Assigned By Ta");
725     String slotId1 =
String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
alueOf(slot.getEndTime()));
726     allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
727
728     studentList.get("Student1").moduleList.put("Module1", slot);
729
730
731     slot = new TimeSlot();

```

```

732     slot.setModuleName("Module2");slot.setGroupId(2);slot.setDate(5);slot.setStartTime(12)
;slot.setEndTime(13);slot.setStatus("Assigned By Ta");
733     slotId1 =
String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
alueOf(slot.getEndTime()));
734     allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
735
736     studentList.get("Student1").moduleList.put("Module2", slot);
737
738     slot = new TimeSlot();
739     slot.setModuleName("Module3");slot.setGroupId(2);slot.setDate(2);slot.setStartTime(12)
;slot.setEndTime(13);slot.setStatus("Assigned By Ta");
740     slotId1 =
String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
alueOf(slot.getEndTime()));
741 // allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
742
743     studentList.get("Student1").moduleList.put("Module3", slot);
744
745
746
747
748     Student student2 = new Student();
749     student2.name = "Student2";
750     student2.moduleList = new HashMap<String,TimeSlot>();
751
752     // add to student list
753     studentList.put("Student2", student2);
754
755     TimeSlot slot2 = new TimeSlot();
756     slot2.setModuleName("Module1");slot2.setGroupId(1);slot2.setDate(1);slot2.setStartTime
(9);slot2.setEndTime(10);slot2.setStatus("Assigned By Ta");
757     String slotId2 =
String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
758     allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
759
760     studentList.get("Student2").moduleList.put("Module1", slot2);
761
762     slot2 = new TimeSlot();
763     slot2.setModuleName("Module2");slot2.setGroupId(1);slot2.setDate(4);slot2.setStartTime
(9);slot2.setEndTime(10);slot2.setStatus("Assigned By Ta");
764     slotId2 =
String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
765     allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
766
767     studentList.get("Student2").moduleList.put("Module2", slot2);
768
769     slot2 = new TimeSlot();
770     slot2.setModuleName("Module3");slot2.setGroupId(1);slot2.setDate(3);slot2.setStartTime
(15);slot2.setEndTime(16);slot2.setStatus("Assigned By Ta");
771     slotId2 =
String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
772 // allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
773
774     studentList.get("Student2").moduleList.put("Module3", slot2);
775 }
776
777
778 private static void Test3ModulesButSwap1() {

```

```

779
780
781     studentList = new HashMap<String, Student>();
782     Student student = new Student();
783     student.name = "Student1";
784     student.moduleList = new HashMap<String, TimeSlot>();
785
786     // add to student list
787     studentList.put("Student1", student);
788
789
790     TimeSlot slot = new TimeSlot();
791     slot.setModuleName("Module1");slot.setGroupId(2);slot.setDate(1);slot.setStartTime(12)
;slot.setEndTime(13);slot.setStatus("Assigned By Ta");
792     String slotId1 =
String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
alueOf(slot.getEndTime()));
793     allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
794
795     studentList.get("Student1").moduleList.put("Module1", slot);
796
797
798     slot = new TimeSlot();
799     slot.setModuleName("Module2");slot.setGroupId(2);slot.setDate(5);slot.setStartTime(12)
;slot.setEndTime(13);slot.setStatus("Assigned By Ta");
800     slotId1 =
String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
alueOf(slot.getEndTime()));
801 // allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
802
803     studentList.get("Student1").moduleList.put("Module2", slot);
804
805     slot = new TimeSlot();
806     slot.setModuleName("Module3");slot.setGroupId(2);slot.setDate(2);slot.setStartTime(12)
;slot.setEndTime(13);slot.setStatus("Assigned By Ta");
807     slotId1 =
String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
alueOf(slot.getEndTime()));
808 // allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
809
810     studentList.get("Student1").moduleList.put("Module3", slot);
811
812
813
814
815     Student student2 = new Student();
816     student2.name = "Student2";
817     student2.moduleList = new HashMap<String, TimeSlot>();
818
819     // add to student list
820     studentList.put("Student2", student2);
821
822     TimeSlot slot2 = new TimeSlot();
823     slot2.setModuleName("Module1");slot2.setGroupId(1);slot2.setDate(1);slot2.setStartTime
(9);slot2.setEndTime(10);slot2.setStatus("Assigned By Ta");
824     String slotId2 =
String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
825     allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
826
827     studentList.get("Student2").moduleList.put("Module1", slot2);
828

```



```

829     slot2 = new TimeSlot();
830     slot2.setModuleName("Module2");slot2.setGroupId(1);slot2.setDate(4);slot2.setStartTime
    (9);slot2.setEndTime(10);slot2.setStatus("Assigned By Ta");
831     slotId2 =
        String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
            .valueOf(slot2.getEndTime()));
832 // allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
833
834     studentList.get("Student2").moduleList.put("Module2", slot2);
835
836     slot2 = new TimeSlot();
837     slot2.setModuleName("Module3");slot2.setGroupId(1);slot2.setDate(3);slot2.setStartTime
    (15);slot2.setEndTime(16);slot2.setStatus("Assigned By Ta");
838     slotId2 =
        String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
            .valueOf(slot2.getEndTime()));
839 // allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
840
841     studentList.get("Student2").moduleList.put("Module3", slot2);
842 }
843
844 private static void Test3ModulesNoSwapNeeded() {
845
846
847     studentList = new HashMap<String, Student>();
848     Student student = new Student();
849     student.name = "Student1";
850     student.moduleList = new HashMap<String,TimeSlot>();
851
852     // add to student list
853     studentList.put("Student1", student);
854
855
856     TimeSlot slot = new TimeSlot();
857     slot.setModuleName("Module1");slot.setGroupId(2);slot.setDate(1);slot.setStartTime(12)
    ;slot.setEndTime(13);slot.setStatus("Assigned By Ta");
858     String slotId1 =
        String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
            alueOf(slot.getEndTime()));
859 // allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
860
861     studentList.get("Student1").moduleList.put("Module1", slot);
862
863
864     slot = new TimeSlot();
865     slot.setModuleName("Module2");slot.setGroupId(2);slot.setDate(5);slot.setStartTime(12)
    ;slot.setEndTime(13);slot.setStatus("Assigned By Ta");
866     slotId1 =
        String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
            alueOf(slot.getEndTime()));
867 // allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
868
869     studentList.get("Student1").moduleList.put("Module2", slot);
870
871     slot = new TimeSlot();
872     slot.setModuleName("Module3");slot.setGroupId(2);slot.setDate(2);slot.setStartTime(12)
    ;slot.setEndTime(13);slot.setStatus("Assigned By Ta");
873     slotId1 =
        String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
            alueOf(slot.getEndTime()));
874 // allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
875

```

Main.java

```

876     studentList.get("Student1").moduleList.put("Module3", slot);
877
878
879
880
881     Student student2 = new Student();
882     student2.name = "Student2";
883     student2.moduleList = new HashMap<String, TimeSlot>();
884
885     // add to student list
886     studentList.put("Student2", student2);
887
888     TimeSlot slot2 = new TimeSlot();
889     slot2.setModuleName("Module1"); slot2.setGroupId(1); slot2.setDate(1); slot2.setStartTime
(9); slot2.setEndTime(10); slot2.setStatus("Assigned By Ta");
890     String slotId2 =
        String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
891 // allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
892
893     studentList.get("Student2").moduleList.put("Module1", slot2);
894
895     slot2 = new TimeSlot();
896     slot2.setModuleName("Module2"); slot2.setGroupId(1); slot2.setDate(4); slot2.setStartTime
(9); slot2.setEndTime(10); slot2.setStatus("Assigned By Ta");
897     slotId2 =
        String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
898 // allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
899
900     studentList.get("Student2").moduleList.put("Module2", slot2);
901
902     slot2 = new TimeSlot();
903     slot2.setModuleName("Module3"); slot2.setGroupId(1); slot2.setDate(3); slot2.setStartTime
(15); slot2.setEndTime(16); slot2.setStatus("Assigned By Ta");
904     slotId2 =
        String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
905 // allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
906
907     studentList.get("Student2").moduleList.put("Module3", slot2);
908 }
909
910 private static void _3Modules_3Agents_only_2SwapSlots() {
911     studentList = new HashMap<String, Student>();
912     Student student = new Student();
913     student.name = "Student1";
914     student.moduleList = new HashMap<String, TimeSlot>();
915
916     // add to student list
917     studentList.put("Student1", student);
918
919
920     TimeSlot slot = new TimeSlot();
921     slot.setModuleName("Module1"); slot.setGroupId(2); slot.setDate(1); slot.setStartTime(12)
; slot.setEndTime(13); slot.setStatus("Assigned By Ta");
922     String slotId1 =
        String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
alueOf(slot.getEndTime()));
923 // allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
924
925     studentList.get("Student1").moduleList.put("Module1", slot);

```



```

926
927
928     slot = new TimeSlot();
929     slot.setModuleName("Module2");slot.setGroupId(2);slot.setDate(5);slot.setStartTime(12)
;slot.setEndTime(13);slot.setStatus("Assigned By Ta");
930     slotId1 =
        String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
alueOf(slot.getEndTime()));
931 // allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
932
933     studentList.get("Student1").moduleList.put("Module2", slot);
934
935     slot = new TimeSlot();
936     slot.setModuleName("Module3");slot.setGroupId(2);slot.setDate(2);slot.setStartTime(12)
;slot.setEndTime(13);slot.setStatus("Assigned By Ta");
937     slotId1 =
        String.valueOf(slot.getDate()).concat(String.valueOf(slot.getStartTime())).concat(String.v
alueOf(slot.getEndTime()));
938 // allPrefs.get(0).replace(slotId1, "outside work or caring commitments");
939
940     studentList.get("Student1").moduleList.put("Module3", slot);
941
942
943
944
945     Student student2 = new Student();
946     student2.name = "Student2";
947     student2.moduleList = new HashMap<String,TimeSlot>();
948
949     // add to student list
950     studentList.put("Student2", student2);
951
952     TimeSlot slot2 = new TimeSlot();
953     slot2.setModuleName("Module1");slot2.setGroupId(1);slot2.setDate(1);slot2.setStartTime
(9);slot2.setEndTime(10);slot2.setStatus("Assigned By Ta");
954     String slotId2 =
        String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
955 // allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
956
957     studentList.get("Student2").moduleList.put("Module1", slot2);
958
959     slot2 = new TimeSlot();
960     slot2.setModuleName("Module2");slot2.setGroupId(2);slot2.setDate(4);slot2.setStartTime
(9);slot2.setEndTime(10);slot2.setStatus("Assigned By Ta");
961     slotId2 =
        String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
962 // allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
963
964     studentList.get("Student2").moduleList.put("Module2", slot2);
965
966     slot2 = new TimeSlot();
967     slot2.setModuleName("Module3");slot2.setGroupId(1);slot2.setDate(3);slot2.setStartTime
(15);slot2.setEndTime(16);slot2.setStatus("Assigned By Ta");
968     slotId2 =
        String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
969     allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
970
971     studentList.get("Student2").moduleList.put("Module3", slot2);
972

```

Main.java

```
973
974
975 Student student3 = new Student();
976 student3.name = "Student2";
977 student3.moduleList = new HashMap<String,TimeSlot>();
978
979 // add to student list
980 studentList.put("Student3", student3);
981
982 slot2 = new TimeSlot();
983 slot2.setModuleName("Module1");slot2.setGroupId(1);slot2.setDate(1);slot2.setStartTime
(9);slot2.setEndTime(10);slot2.setStatus("Assigned By Ta");
984 slotId2 =
String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
985 // allPrefs.get(1).replace(slotId2, "outside work or caring commitments");
986
987 studentList.get("Student3").moduleList.put("Module1", slot2);
988
989 slot2 = new TimeSlot();
990 slot2.setModuleName("Module2");slot2.setGroupId(1);slot2.setDate(4);slot2.setStartTime
(9);slot2.setEndTime(10);slot2.setStatus("Assigned By Ta");
991 slotId2 =
String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
992 // allPrefs.get(2).replace(slotId2, "outside work or caring commitments");
993
994 studentList.get("Student3").moduleList.put("Module2", slot2);
995
996 slot2 = new TimeSlot();
997 slot2.setModuleName("Module3");slot2.setGroupId(2);slot2.setDate(5);slot2.setStartTime
(15);slot2.setEndTime(16);slot2.setStatus("Assigned By Ta");
998 slotId2 =
String.valueOf(slot2.getDate()).concat(String.valueOf(slot2.getStartTime())).concat(String
.valueOf(slot2.getEndTime()));
999 allPrefs.get(2).replace(slotId2, "outside work or caring commitments");
1000
1001 studentList.get("Student3").moduleList.put("Module3", slot2);
1002 }
1003 }
1004
1005
1006
1007
1008
```