Tung Leong Kok 53028089  
  
Schedule.java

From

|  |
| --- |
| String inputFile = “CS3343\_data2.txt”; |

To

|  |
| --- |
| // String inputFile = “CS3343\_data2.txt”;  String inputFile = args[0]; |

Schedule.java

From

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | System.out.println("                                       |-------------------------|                                       "); | | System.out.println("                                       |   Visualized timetable  |                                       "); | | System.out.println("|------------|------------|------------|------------|------------|------------|------------|------------|"); | | System.out.println("|Time        |Monday      |Tuesday     |Wednesday   |Thursday    |Friday      |Saturday    |Sunday      |"); | |

To

|  |
| --- |
| System.out.println(printScheduleHeader()); |

Schedule.java

Add

|  |
| --- |
| public static String printScheduleHeader()  {  return(" |-------------------------| \n | Visualized timetable | \n|------------|------------|------------|------------|------------|------------|------------|------------|\n|Time |Monday |Tuesday |Wednesday |Thursday |Friday |Saturday |Sunday |");  } |

TestSchedule.java  
  
Add

|  |
| --- |
| import java.io.\*; |

Add

|  |
| --- |
| private final ByteArrayOutputStream outContent = new ByteArrayOutputStream(); |

From

|  |
| --- |
| **public** **void** setUp() { timeslots = **new** ArrayList<Timeslot>(); } |

To

|  |
| --- |
| **public** **void** setUp() { timeslots = **new** ArrayList<Timeslot>(); System.setOut(**new** PrintStream(outContent)); } |

From

|  |
| --- |
| public void tearDown() {} |

To

|  |
| --- |
| public void tearDown() {System.setOut(null);} |

From

|  |
| --- |
| public void testExtractByCode() {  Schedule.readTimeslots(timeslots, "CS3343\_data.txt");  String expected = "AC2";  ArrayList<Timeslot> t = Schedule.extractTimeslotsByCode(timeslots, expected);  boolean result = true;  for (Timeslot i : t)  if (!i.getBuilding().equals(expected))  result = false;  assertEquals(result, true);  } |

To

|  |
| --- |
| public void testExtractByCode() {  Timeslot a = new Timeslot("40001","CS3332","C01", "AC1", "LT-1", 14, 16, Weekday.Mon.getDay());  Timeslot b = new Timeslot("40002","CS2332","LA1", "AC2", "5503", 13, 16, Weekday.Tue.getDay());  Timeslot c = new Timeslot("40003","CS3301","LA1", "AC1", "LT-3", 9, 11.5, Weekday.Tue.getDay());  Timeslot d = new Timeslot("40004","CS3201","CA1", "AC3", "6208", 10, 12, Weekday.Tue.getDay());  Timeslot e = new Timeslot("40005","CS3443","CB1", "AC1", "LT-2", 12, 16, Weekday.Tue.getDay());  Timeslot f = new Timeslot("40006","CS3443","CA1", "AC1", "LT-2", 18, 22, Weekday.Tue.getDay());  timeslots.add(a);  timeslots.add(b);  timeslots.add(c);  timeslots.add(d);  timeslots.add(e);  timeslots.add(f);  String expected = "CS3443";  ArrayList<Timeslot> t = Schedule.extractTimeslotsByCode(timeslots, expected);  assertEquals(t.toString().equals("[CS3443-CB1, CS3443-CA1]"), true);  } |

From

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | public void testExtractByCodeFalse() { | | Timeslot a = new Timeslot("40001","CS3332","C01", "AC1", "LT-1", 14, 16, Weekday.Mon.getDay()); | | Timeslot b = new Timeslot("40002","CS2332","LA1", "AC2", "5503", 13, 16, Weekday.Tue.getDay()); | | Timeslot c = new Timeslot("40003","CS3301","LA1", "AC1", "LT-3", 9, 11.5, Weekday.Tue.getDay()); | | Timeslot d = new Timeslot("40004","CS3201","CA1", "AC3", "6208", 10, 12, Weekday.Tue.getDay()); | | Timeslot e = new Timeslot("40005","CS3443","CB1", "AC1", "LT-2", 12, 16, Weekday.Tue.getDay()); | | timeslots.add(a); | | timeslots.add(b); | | timeslots.add(c); | | timeslots.add(d); | | timeslots.add(e); | | String expected = "AC2"; | | boolean result = true; | | for (Timeslot i : timeslots) | | if (!i.getBuilding().equals(expected)) | | result = false; | | assertEquals(result, false); | | } | |

To

|  |
| --- |
| public void testExtractByCode2() {  Timeslot a = new Timeslot("40001","CS3332","C01", "AC1", "LT-1", 14, 16, Weekday.Mon.getDay());  Timeslot b = new Timeslot("40002","CS2332","LA1", "AC2", "5503", 13, 16, Weekday.Tue.getDay());  Timeslot c = new Timeslot("40003","CS3301","LA1", "AC1", "LT-3", 9, 11.5, Weekday.Tue.getDay());  Timeslot d = new Timeslot("40004","CS3201","CA1", "AC3", "6208", 10, 12, Weekday.Tue.getDay());  Timeslot e = new Timeslot("40005","CS3443","CB1", "AC1", "LT-2", 12, 16, Weekday.Tue.getDay());  Timeslot f = new Timeslot("40006","CS3443","CA1", "AC1", "LT-2", 18, 22, Weekday.Tue.getDay());  timeslots.add(a);  timeslots.add(b);  timeslots.add(c);  timeslots.add(d);  timeslots.add(e);  timeslots.add(f);  String expected = "CS3332";  ArrayList<Timeslot> t = Schedule.extractTimeslotsByCode(timeslots, expected);  assertEquals(t.toString().equals("[CS3332-C01]"), true);  } |

Add

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | public void testToString2() { | | Timeslot a = new Timeslot("40001","CS3332","C01", "AC1", "LT-1", 14, 16, Weekday.Mon.getDay()); | | Timeslot b = new Timeslot("40002","CS2332","LA1", "AC2", "5503", 13, 16, Weekday.Tue.getDay()); | | Timeslot c = new Timeslot("40003","CS3301","LA1", "AC1", "LT-3", 9, 11.5, Weekday.Wed.getDay()); | | Timeslot d = new Timeslot("40004","CS3201","CA1", "AC3", "6208", 10, 12, Weekday.Tue.getDay()); | | Timeslot e = new Timeslot("40005","CS3443","CB1", "AC1", "LT-2", 12, 16, Weekday.Tue.getDay()); | |  | | timeslots.add(a); | | timeslots.add(b); | |  | | assertEquals(timeslots.toString().equals("[CS3332-C01, CS2332-LA1]"), true); | |  | | timeslots.clear(); | | timeslots.add(c); | | timeslots.add(d); | | timeslots.add(e); | |  | | assertEquals(timeslots.toString().equals("[CS3301-LA1, CS3201-CA1, CS3443-CB1]"), true); | |  | | } | |  | | // Test case 20: Test allCourses | | @Test | | public void testAllCourses() { | | Timeslot a = new Timeslot("40001","CS3332","C01", "AC1", "LT-1", 14, 16, Weekday.Mon.getDay()); | | Timeslot b = new Timeslot("40002","CS2332","LA1", "AC2", "5503", 13, 16, Weekday.Tue.getDay()); | | Timeslot c = new Timeslot("40003","CS3301","LA1", "AC1", "LT-3", 9, 11.5, Weekday.Wed.getDay()); | | Timeslot d = new Timeslot("40004","CS3201","CA1", "AC3", "6208", 10, 12, Weekday.Tue.getDay()); | | Timeslot e = new Timeslot("40005","CS3443","CB1", "AC1", "LT-2", 12, 16, Weekday.Tue.getDay()); | | Timeslot f = new Timeslot("40006","CS3443","CA1", "AC1", "LT-2", 18, 22, Weekday.Tue.getDay()); | |  | | timeslots.add(a); | | timeslots.add(b); | | timeslots.add(c); | | timeslots.add(d); | | timeslots.add(e); | | timeslots.add(f); | |  | | ArrayList<String> uniqueCourses = Schedule.allCourses(timeslots); | | assertEquals(uniqueCourses.toString().equals("[CS3332, CS2332, CS3301, CS3201, CS3443]"), true); | |  | | } | |  | | // Test case 21: Test permutate | | @Test | | public void testPermutate() { | | Timeslot a = new Timeslot("40001","CS3332","C01", "AC1", "LT-1", 14, 16, Weekday.Mon.getDay()); | | Timeslot b = new Timeslot("40002","CS2332","LA1", "AC2", "5503", 13, 16, Weekday.Tue.getDay()); | | Timeslot c = new Timeslot("40003","CS3301","LA1", "AC1", "LT-3", 9, 11.5, Weekday.Wed.getDay()); | | Timeslot d = new Timeslot("40004","CS3201","CA1", "AC3", "6208", 10, 12, Weekday.Tue.getDay()); | | Timeslot e = new Timeslot("40005","CS3443","CB1", "AC1", "LT-2", 12, 16, Weekday.Tue.getDay()); | | Timeslot f = new Timeslot("40006","CS3443","CA1", "AC1", "LT-2", 18, 22, Weekday.Tue.getDay()); | |  | | timeslots.add(a); | | timeslots.add(b); | | timeslots.add(c); | |  | | ArrayList<Timeslot> timeslots2 = new ArrayList<Timeslot>(); | |  | | timeslots2.add(d); | | timeslots2.add(e); | | timeslots2.add(f); | |  | | assertEquals(Schedule.permutate(timeslots, timeslots2).toString().equals("[[CS3332-C01, CS3201-CA1], [CS3332-C01, CS3443-CB1], [CS3332-C01, CS3443-CA1], [CS2332-LA1, CS3201-CA1], [CS2332-LA1, CS3443-CB1], [CS2332-LA1, CS3443-CA1], [CS3301-LA1, CS3201-CA1], [CS3301-LA1, CS3443-CB1], [CS3301-LA1, CS3443-CA1]]"), true); | |  | | } | |  | | // Test case 22: Test permutateArrayList | | @Test | | public void testPermutateArrayList() { | | Timeslot a = new Timeslot("40001","CS3332","C01", "AC1", "LT-1", 14, 16, Weekday.Mon.getDay()); | | Timeslot b = new Timeslot("40002","CS3332","LA1", "AC2", "5503", 13, 16, Weekday.Tue.getDay()); | | Timeslot c = new Timeslot("40003","CS3332","LA2", "AC1", "LT-3", 9, 12, Weekday.Wed.getDay()); | | Timeslot d = new Timeslot("40004","CS3443","LA2", "AC3", "6208", 10, 12, Weekday.Tue.getDay()); | | Timeslot e = new Timeslot("40005","CS3443","LA1", "AC1", "LT-2", 12, 13, Weekday.Tue.getDay()); | | Timeslot f = new Timeslot("40006","CS3443","C01", "AC1", "LT-2", 18, 22, Weekday.Tue.getDay()); | |  | | ArrayList<ArrayList<Timeslot>> timeslotsA = new ArrayList<ArrayList<Timeslot>>(); | | ArrayList<Timeslot> timeslotsA1 = new ArrayList<Timeslot>(); | | ArrayList<Timeslot> timeslotsA2 = new ArrayList<Timeslot>(); | |  | | timeslotsA1.add(a); | | timeslotsA1.add(b); | | timeslotsA2.add(a); | | timeslotsA2.add(c); | | timeslotsA.add(timeslotsA1); | | timeslotsA.add(timeslotsA2); | |  | | ArrayList<ArrayList<Timeslot>> timeslotsB = new ArrayList<ArrayList<Timeslot>>(); | | ArrayList<Timeslot> timeslotsB1 = new ArrayList<Timeslot>(); | | ArrayList<Timeslot> timeslotsB2 = new ArrayList<Timeslot>(); | |  | | timeslotsB1.add(d); | | timeslotsB1.add(f); | | timeslotsB2.add(e); | | timeslotsB2.add(f); | | timeslotsB.add(timeslotsB1); | | timeslotsB.add(timeslotsB2); | |  | | ArrayList<ArrayList<Timeslot>> permutated = Schedule.permutateArrayList(timeslotsA, timeslotsB); | |  | | ArrayList<ArrayList<Timeslot>> timeslotsAexpected = new ArrayList<ArrayList<Timeslot>>(); | | ArrayList<Timeslot> timeslotsExp1 = new ArrayList<Timeslot>(); | | ArrayList<Timeslot> timeslotsExp2 = new ArrayList<Timeslot>(); | | ArrayList<Timeslot> timeslotsExp3 = new ArrayList<Timeslot>(); | | ArrayList<Timeslot> timeslotsExp4 = new ArrayList<Timeslot>(); | |  | | timeslotsExp1.add(a); | | timeslotsExp1.add(b); | | timeslotsExp1.add(d); | | timeslotsExp1.add(f); | | timeslotsExp2.add(a); | | timeslotsExp2.add(b); | | timeslotsExp2.add(e); | | timeslotsExp2.add(f); | | timeslotsExp3.add(a); | | timeslotsExp3.add(c); | | timeslotsExp3.add(d); | | timeslotsExp3.add(f); | | timeslotsExp4.add(a); | | timeslotsExp4.add(c); | | timeslotsExp4.add(e); | | timeslotsExp4.add(f); | |  | | timeslotsAexpected.add(timeslotsExp1); | | timeslotsAexpected.add(timeslotsExp2); | | timeslotsAexpected.add(timeslotsExp3); | | timeslotsAexpected.add(timeslotsExp4); | |  | | assertEquals(permutated.equals(timeslotsAexpected), true); | |  | | } | |  | | // Test case 22: Test permutateArrayList | | @Test | | public void testGeneratePermutations() { | | Timeslot a = new Timeslot("40001","CS3332","C01", "AC1", "LT-1", 14, 16, Weekday.Mon.getDay()); | | Timeslot b = new Timeslot("40002","CS3332","LA1", "AC2", "5503", 13, 16, Weekday.Tue.getDay()); | | Timeslot c = new Timeslot("40003","CS3332","LA2", "AC1", "LT-3", 9, 12, Weekday.Wed.getDay()); | | Timeslot d = new Timeslot("40004","CS3443","LA2", "AC3", "6208", 10, 12, Weekday.Tue.getDay()); | | Timeslot e = new Timeslot("40005","CS3443","LA1", "AC1", "LT-2", 12, 13, Weekday.Tue.getDay()); | | Timeslot f = new Timeslot("40006","CS3443","C01", "AC1", "LT-2", 18, 22, Weekday.Tue.getDay()); | | Timeslot g = new Timeslot("40007","CS3201","LA1", "MMW", "2603", 9, 11.5, Weekday.Thu.getDay()); | | Timeslot h = new Timeslot("40008","CS3201","C01", "AC1", "LT-17", 10, 12, Weekday.Thu.getDay()); | |  | | timeslots.add(a); | | timeslots.add(b); | | timeslots.add(c); | | timeslots.add(d); | | timeslots.add(e); | | timeslots.add(f); | | timeslots.add(g); | | timeslots.add(h); | |  | | ArrayList<Timeslot> CS3332 = Schedule.extractTimeslotsByCode(timeslots, "CS3332"); | | ArrayList<Timeslot> CS3332l = Schedule.extractTimeslotsByType(CS3332, "Lecture"); | | ArrayList<Timeslot> CS3332t = Schedule.extractTimeslotsByType(CS3332, "Tutorial"); | |  | | ArrayList<Timeslot> CS3443 = Schedule.extractTimeslotsByCode(timeslots, "CS3443"); | | ArrayList<Timeslot> CS3443l = Schedule.extractTimeslotsByType(CS3443, "Lecture"); | | ArrayList<Timeslot> CS3443t = Schedule.extractTimeslotsByType(CS3443, "Tutorial"); | |  | | ArrayList<Timeslot> CS3201 = Schedule.extractTimeslotsByCode(timeslots, "CS3201"); | | ArrayList<Timeslot> CS3201l = Schedule.extractTimeslotsByType(CS3201, "Lecture"); | | ArrayList<Timeslot> CS3201t = Schedule.extractTimeslotsByType(CS3201, "Tutorial"); | |  | | ArrayList<ArrayList<Timeslot>> CS3332p = Schedule.permutate(CS3332l, CS3332t); | | ArrayList<ArrayList<Timeslot>> CS3443p = Schedule.permutate(CS3443l, CS3443t); | | ArrayList<ArrayList<Timeslot>> CS3201p = Schedule.permutate(CS3201l, CS3201t); | |  | | ArrayList<ArrayList<ArrayList<Timeslot>>> allCourses = new ArrayList<ArrayList<ArrayList<Timeslot>>>(); | | allCourses.add(CS3332p); | | allCourses.add(CS3443p); | | allCourses.add(CS3201p); | |  | | ArrayList<ArrayList<ArrayList<Timeslot>>> permutated = Schedule.GeneratePermutations(allCourses); | | //System.out.println(permutated); | |  | | ArrayList<ArrayList<ArrayList<Timeslot>>> expected = new ArrayList<ArrayList<ArrayList<Timeslot>>>(); | | ArrayList<ArrayList<Timeslot>> expectedT = new ArrayList<ArrayList<Timeslot>>(); | | ArrayList<Timeslot> expected1 = new ArrayList<Timeslot>(); | | ArrayList<Timeslot> expected2 = new ArrayList<Timeslot>(); | | ArrayList<Timeslot> expected3 = new ArrayList<Timeslot>(); | | ArrayList<Timeslot> expected4 = new ArrayList<Timeslot>(); | |  | | expected1.add(h); | | expected1.add(g); | | expected1.add(a); | | expected1.add(b); | | expected1.add(f); | | expected1.add(d); | |  | | expected2.add(h); | | expected2.add(g); | | expected2.add(a); | | expected2.add(b); | | expected2.add(f); | | expected2.add(e); | |  | | expected3.add(h); | | expected3.add(g); | | expected3.add(a); | | expected3.add(c); | | expected3.add(f); | | expected3.add(d); | |  | | expected4.add(h); | | expected4.add(g); | | expected4.add(a); | | expected4.add(c); | | expected4.add(f); | | expected4.add(e); | |  | | expectedT.add(expected1); | | expectedT.add(expected2); | | expectedT.add(expected3); | | expectedT.add(expected4); | |  | | expected.add(expectedT); | |  | | assertEquals(permutated.equals(expected), true); | | } | |  | | // Test case 23: Test printScheduleHeader | | @Test | | public void testPrintScheduleHeader() { | | String x = Schedule.printScheduleHeader(); | | String expected = "                                       |-------------------------|                                       \n                                       |   Visualized timetable  |                                       \n|------------|------------|------------|------------|------------|------------|------------|------------|\n|Time        |Monday      |Tuesday     |Wednesday   |Thursday    |Friday      |Saturday    |Sunday      |"; | | assertEquals(x.equals(expected), true);   |  | | --- | | // Test case 24: Test printSchedule | | @Test | | public void testPrintSchedule() { | | Timeslot a = new Timeslot("40001","CS3332","C01", "AC1", "LT-1", 14, 16, Weekday.Mon.getDay()); | | Timeslot b = new Timeslot("40002","CS2332","LA1", "AC2", "5503", 13, 16, Weekday.Tue.getDay()); | | Timeslot c = new Timeslot("40003","CS3301","LA1", "AC1", "LT-3", 9, 11, Weekday.Wed.getDay()); | | Timeslot d = new Timeslot("40004","CS3201","CA1", "AC3", "6208", 10, 12, Weekday.Tue.getDay()); | | Timeslot f = new Timeslot("40006","CS3443","CA1", "AC1", "LT-2", 18, 22, Weekday.Tue.getDay()); | |  | | timeslots.add(a); | | timeslots.add(b); | | timeslots.add(c); | | timeslots.add(d); | | timeslots.add(f); | | Schedule.printSchedule(timeslots); | |  | | String expected = "                                       |-------------------------|                                       \n" + | | "                                       |   Visualized timetable  |                                       \n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\n" + | | "|Time        |Monday      |Tuesday     |Wednesday   |Thursday    |Friday      |Saturday    |Sunday      |\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\n" + | | "|0800-0850   |            |            |            |            |            |            |            |\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\n" + | | "|0900-0950   |            |            |CS3301-LA1  |            |            |            |            |\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\n" + | | "|1000-1050   |            |CS3201-CA1  |CS3301-LA1  |            |            |            |            |\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\n" + | | "|1100-1150   |            |CS3201-CA1  |            |            |            |            |            |\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\n" + | | "|1200-1250   |            |            |            |            |            |            |            |\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\n" + | | "|1300-1350   |            |CS2332-LA1  |            |            |            |            |            |\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\n" + | | "|1400-1450   |CS3332-C01  |CS2332-LA1  |            |            |            |            |            |\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\n" + | | "|1500-1550   |CS3332-C01  |CS2332-LA1  |            |            |            |            |            |\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\n" + | | "|1600-1650   |            |            |            |            |            |            |            |\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\n" + | | "|1700-1750   |            |            |            |            |            |            |            |\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\n" + | | "|1800-1850   |            |CS3443-CA1  |            |            |            |            |            |\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\n" + | | "|1900-1950   |            |CS3443-CA1  |            |            |            |            |            |\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\n" + | | "|2000-2050   |            |CS3443-CA1  |            |            |            |            |            |\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\n" + | | "|2100-2150   |            |CS3443-CA1  |            |            |            |            |            |\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\n" + | | "|2200-2250   |            |            |            |            |            |            |            |\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|" + | | ""; | |  | | assertEquals(expected.replaceAll("\n", ""), outContent.toString().replaceAll("\r\n", "").replaceAll("\n", "")); | |  | | //assertEquals(x.equals(expected), true); | |  | | } | |  | | // Test case 25: Test main | | @Test | | public void testMain() { | | String[] args = {"CS3343\_data2.txt"}; | | Schedule.main(args); | | String expected = "There are 4248 possible combinations.\r\n" + | | "                                       |-------------------------|                                       \r\n" + | | "                                       |   Visualized timetable  |                                       \r\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\r\n" + | | "|Time        |Monday      |Tuesday     |Wednesday   |Thursday    |Friday      |Saturday    |Sunday      |\r\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\r\n" + | | "|0800-0850   |            |            |            |            |            |            |            |\r\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\r\n" + | | "|0900-0950   |            |            |            |            |            |            |            |\r\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\r\n" + | | "|1000-1050   |            |CS2010-C0A  |CS2201-C0A  |            |            |            |            |\r\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\r\n" + | | "|1100-1150   |CS3332-T0A  |CS2010-C0A  |CS2201-C0A  |            |            |            |            |\r\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\r\n" + | | "|1200-1250   |            |            |CS2205-T01  |            |            |            |            |\r\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\r\n" + | | "|1300-1350   |CS3332-C0A  |CS2010-T0A  |CS2201-T0A  |            |            |            |            |\r\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\r\n" + | | "|1400-1450   |CS3332-C0A  |            |CS2205-C0A  |            |            |            |            |\r\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\r\n" + | | "|1500-1550   |CS3332-C0A  |            |            |CS4321-C01  |CS2332-L0A  |            |            |\r\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\r\n" + | | "|1600-1650   |CS2112-C0A  |            |            |CS4321-C01  |CS2332-L0A  |            |            |\r\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\r\n" + | | "|1700-1750   |CS2112-C0A  |            |            |            |CS2332-C0A  |            |            |\r\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\r\n" + | | "|1800-1850   |CS2112-T0A  |            |CS4321-T0A  |            |            |            |            |\r\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\r\n" + | | "|1900-1950   |            |            |CS4321-T0A  |            |            |            |            |\r\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\r\n" + | | "|2000-2050   |            |            |            |            |            |            |            |\r\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\r\n" + | | "|2100-2150   |            |            |            |            |            |            |            |\r\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\r\n" + | | "|2200-2250   |            |            |            |            |            |            |            |\r\n" + | | "|------------|------------|------------|------------|------------|------------|------------|------------|\r\n" + | | ""; | | assertEquals(expected.replaceAll("\r\n", ""), outContent.toString().replaceAll("\r\n", "").replaceAll("\n", "")); | |  | | } | |  | | // Test case 26: Test printSchedule(No possible combination) | | @Test | | public void testPrintScheduleNo() { | | String[] args = {"CS3343\_data.txt"}; | | Schedule.main(args); | | String expected = "There is no possible combination i.e. You should remove at least 1 course."; | | assertEquals(expected.replaceAll("\r\n", ""), outContent.toString().replaceAll("\r\n", "").replaceAll("\n", "")); | | } | | } | | |