**SEAT USER GUIDE (For SEAT v0.2.1)**

This document explains how to use the SEAT software. Please read this document thoroughly before using the SEAT software. This document does not contain the Code documentation. Look at the SEAT Code Documentation document for that.

All changes to be preferably made in the original Google Document, and not the local Microsoft Word version uploaded on Github.

Note that this Guide and the SEAT software is not to be distributed or used without explicit permission from the SEAT team, IIT Madras.

**SEAT Software Team:**

1. Akshay Utture - [akshay.utture@gmail.com](mailto:akshay.utture@gmail.com)
2. Meghana Nasre - [meghana.nasre@gmail.com](mailto:meghana.nasre@gmail.com)
3. Saranya Sundaresan Malarvizhi - [saranyahighflier@gmail.com](mailto:saranyahighflier@gmail.com)
4. John Augustine - [j.e.augustine@gmail.com](mailto:j.e.augustine@gmail.com)
5. Vedant Somani - [iamvedantsomani@gmail.com](mailto:iamvedantsomani@gmail.com)

**Document Version:**

This version of the document was prepared in July 2017, for the SEAT v0.2.1, which was used to make the Jul-Nov 2017 SEAT allotment (First Round).

**INDEX**

1) What does this software do?

2) Important Terminology Used

3) Input Files used by the Software

4) Output Files given out by the Software

5) Sample Input and Output Files

6) Running the software from the terminal

7) Running the software from the GUI

8 ) Modifications to Workflow data to fit it to the Software

**1) What does this software do?**

After the curriculum change at IIT Madras for the 2015 Btech/DD students onwards, the paper based COT (Consent of teacher) system for registering for Elective courses was decided to be replaced with a software tool, in which students would give a preference list for courses which they would like to study. They would be allotted a set of courses as high as possible on their preference list, while respecting constraints like course capacity limits, student credit limits, course slot clashes, etc. This software is called SEAT (Student Elective Allocation tool), and is used to carry out this allotment of courses.

Currently the software is used for allocating Btech and Dual Degree students their Elective courses at IIT Madras. The software can be extended to include other students as well.

The software is designed to read a set of inputs (including student preference lists, course data, slot timings, etc.) and output an allotment, as well as a set of statistics on the allotment in order to quantify how good the allotment is.

**2) Important Terminology Used**

This section explains some of the terminology that will be used through the rest of this document.

1. **Ranking Criteria (for a course’s preference list) :**

A Ranking Criteria will be used by the Course to Rank the students. This ranking of students will be used to generate the course’s preference list if need be. Currently there are 3 ranking criteria

i) Random : The course ranks all students in a random order

ii) Stratified Random : The course first ranks all those students who rank this course as their first choice. Among those that students who ranked this course as their first choice, ties are broken randomly. Next the course ranks all those students who rank this course as their second choice, and break ties within them randomly. This keeps going on until all students who have applied for the course have been ranked by the course.

iii) CGPA : The course ranks students based on their CGPA, and breaks ties randomly.

1. **Colour Code for Courses (for a student’s preference list):**

A student’s preference list is a single list containing all the courses he/she wants to give a preference for, including HS courses, MA courses, department electives,etc. SEAT provides the student with an option to colour code his/her preferences when giving their preferences. Among all the courses coloured with a single colour, at most one course will be allotted. In the student preference file, the colour is represented by a number - each colour corresponds to a number. For example, if a student wishes to provide several Math department courses in his preference list, but wishes to be allotted only 1 of them, then he can colour all his Math department courses with the same colour to ensure that he gets allotted at most 1 of those Math courses.

There is one exception to this rule. The colour represented by the number ‘0’ is a no colour option. Hence there is no restriction on how many courses with the colour ‘0’ can be allotted. So for example, if a student uses the colour 0 for courses CS4000 and CS5000, he could potentially be allocated both the courses

**c) Max Credit Limit provided by department**

There are some departments which specify a maximum credit limit other than the default (currently 60) for some of their students. For example, the CS department allows it’s 5th semester students to opt for a maximum of 63 credits instead of the default (currently 60). This kind of information is provided in this file. Those departments which prescribe the default credit limit need not be mentioned here. Also, note that students are allowed to specify their ‘Max Credits’ in the Student List which is lower than the Max Credit Limit given by their department since the Max Credit Limit is only an upper bound the ‘Max Credit’ value a student provides.

**d) Inside Department and Outside Department version of the course**

Each course has an inside department capacity, which is meant only for students who it considers inside department, and an outside department capacity, which is mainly meant for outside department students, but can also be filled with inside department students. So each course (eg. CS5001) is split into it’s 2 versions - the inside department version (CS5001$inside), and it’s outside department version (CS5001$outside).

By default if the first 2 letters of a student’s roll number (eg. CS13B031) match the department code provided for a Course in the Course List file (eg. CS), then the student is considered as an inside department student. But one can specify special cases where a course may consider students as inside department, even though the 2 letters do not match. These special cases are provided in the ‘Inside Department Specification File’.

Important Examples - these should be added to the file:

* All OE department courses consider all NA students as inside department
* All BT courses consider all BS and BE students as inside department.
* Some EE courses might consider some EP students as inside department
* All PH courses consider EP students as inside department

(Read the next paragraph only when you are familiar with the SEAT system. It is a complex idea)

Sometimes this Inside Department Specification can be used to reserve HS or MA courses for the students for whom an HS or MA course is recommended. For example -

If the AE16 and BE15 batches are the only batches that have a mandated HS electives, we can reserve some popular HS courses like HS3002, HS1090 for them in the following way. Declare AE16 and BE15 batch students as inside department students for these 2 courses. Make the entire capacity of HS3002 and HS1090 as inside department, and make the outside department capacity as 0. This will ensure that only AE16 and BE15 batch students become eligible to allotted to this course. All the other students apply to the outside department capacity of these courses which is 0, and hence cannot be allotted. Remember that by doing this, HS3002 and HS1090 will become unavailable to other students even if there are vacancies. So ensure that all the reserved capacity gets used up. You can even reserve a particular number of seats in the class (instead of the whole class) by the same technique.

**e) High Priority Students**

Some students are considered as High Priority by some course. For example, if the 2nd year CS department students have a mandated HS elective in the F slot, these students can be declared as high priority for all the HS courses in the F slot. The High Priority Students list will contain all such cases of High Priority students

**f) Effective Average Rank**

This is a measure used to quantify how good the allotment is for a student based on the preference numbers he gave in his preference list.

Effective Average Rank = (Sum of Allotted Reduced Ranks)/(Number of Courses allotted)^2

Where

Reduced Rank = (Actual Rank in the Preference list) minus (The number of courses above this preference which were removed due to some conflict - colour or slot or credits)

Example : If a student s1’s preference list is : (c1,c2,c3,c4,c5), and the student is allotted courses (c1,c4). If c2 and c1 conflict (due to slot or colour or the sum of their credits exceeds the credit limit), then the Reduced Ranks are as follows

C1 - 1

C2 - (removed due to conflict with allotted course c1)

C3 - 2

C4 - 3

C5 - 4

So now the Sum of Allotted Reduced ranks = Reduced Rank of c1 + Reduced Rank of c4

= 1 + 3 = 4

Since number of allotted courses is 2, Effective Average Rank = 4 / 2^2 = 4/4 = 1

The reason for using the Reduced Ranks is that if a student has giventhe same colour for his first 10 courses, and he gets allotted his 1st and 11th choice, then the Sum of Ranks is 1+10=11, but the Sum of Reduced Ranks = 1 + 2 = 3, which is a more accurate representation of the allotment. We did allot the student his first and second preference.

Note that while computing the Reduced Ranks we only remove those courses which conflict with a an allotted course which is higher up the preference list. If it conflicts with a course lower down the preference list, it does not get removed. For example, if s2’s preference list is (c1,c2,c3), and the student is allotted c3, and c2-c3 slot conflict, and c1-c2 colour conflict then the Reduced Ranks are

C1 - 1

C2 - 2

C3 - 3

Effective Average Rank = 3 / 1^2 = 3

Note that c2-c3 slot conflict but c3 is below c2, and hence c2 is still counted. c2-c1 colour conflict, but c1 was not allotted, and hence c2 is still counted.

The reason for using a square in the denominator is for normalization is as follows. If a student is allotted his/her first 5 choices, the Sum of Allotted Reduced Ranks is 1+2+3+4+5=15. If we divide by the ‘Number of Courses’, we get 15/5 = 3. But that sounds like we did not do such a good allotment. But in reality, we allotted the student the best set of possible courses he could get. Hence if we divide by ‘Number of Courses^2’ we get 15/25 = 0.6 which is a more accurate representation of how good the allotment was. Allotting the top 5 consecutive courses should get a rank of less than 1.

The overall Effective Average Rank is the arithmetic mean of the Effective Average Ranks taken over all the students

**g) Credit Satisfaction Ratio**

This is a measure used to quantify how many of a student’s required credits are satisfied. It is computed per student. It is simply computed as

Credit Satisfaction Ratio for a student = Number of credits allotted / Number of credits asked

Note that the number of credits being mentioned here is the number of elective credits, not total credits. The ‘number of credits asked’ is computed by subtracting the core course credits from the ‘Max Credits’ provided.

Students not allotted any course are given a Credit Satisfaction Ratio of 1 by default.

The overall Credit Satisfaction Ratio is the arithmetic mean taken over all the students

**3) Input Files used by the Software**

Currently the software takes the following input files - all in a CSV (comma separated values) format. The format of the files is also specified. (The first row of every file is a header line - meant for the header of every column).

1. **Student List File (each row represents 1 student)**
   1. Column 1 : Student’s Roll Number (eg. CS13B031)
   2. Column 2 : Student’s CGPA upto the current semester
   3. Column 3 : Maximum Credits for the upcoming semester. This includes the credits for both Core and Elective Courses. Currently the default value is 60.
2. **Course List File (each row represents 1 course)**
   1. Column 1 : Course Number (eg. CS6100)
   2. Column 2 : 2 letter Department Code (eg. CS)
   3. Column 3 : Maximum students that SEAT can allot to the course. Note that this is different from the maximum Classroom size, since it only includes those students who will be allotted by SEAT. This maximum number does not account for those students who will be allotted from outside SEAT (including students taking the course as a Core Course, MS students, Mtech students, PhD students)
   4. Column 4 : Maximum Outside department students that can be allotted to this course. The entry in Column 3 specifies the total students that SEAT can allocate to the course, and it includes both inside, as well as outside department students. The column 4 entry specifies the limit on only the number of outside department students. The column 4 entry must be lesser than or equal to the column 3 entry.
   5. Column 5 : Ranking Criteria used by the Course to Rank the students. Currently there are 3 ranking criteria - Random (represented by the number 1), Stratified Random (represented by the number 2), and CGPA (represented by the number 3). So this column takes an integer input corresponding to the ranking criteria to be used. The explanation for each Ranking Criterion is given in the section on ‘Important Terminology Used’
   6. Column 6 onwards : Column 6 onwards is a comma separated list of Slots that the course will run in. Note that for each slot mentioned here, there must a corresponding entry in the Slot Timings input file which specifies the timings that the Slot runs in. SLOTS MUST BE TAKEN FROM THE SLOTBOOK DATA. THEY MUST NOT BE TAKEN FROM THE STUDENT PREFERENCE LIST BECAUSE THE SLOT COLUMN HERE COULD BE OUTDATED AND IT HAS SPACE TO MENTION ONLY 1 SLOT. SLOTBOOK DATA IS UP TO DATE AND HAS A COLUMN FOR ADDITIONAL SLOTS
3. **Student Preference List File (each row represents 1 Preference. I.e. 1 student-course pair)**
   1. Column 1 : Student’s Roll Number (eg. CS13B031)
   2. Column 2 : Course Number for the course that the student is providing a preference for
   3. Column 3 : Colour code for this preference of the student. It will be an integer (a unique integer for each unique colour). An explanation on what colour codes are is given in the section on ‘Important Terminology Used’
   4. Column 4 : Type of preference. This tells you if the student is applying for this course as a core course or an Elective course. There is a 3rd entry called Backlog course, which is treated the same as a Core course. Hence only one of these 3 entries are permitted in this column - CORE, ELEC, BACKLOG.
   5. Column 5 : The preference number for this course. This gives the rank the student assigns to the course in his preference list. It only ranks the Elective courses, and has the value of NULL for Core and Backlog courses, since we will not be allotting those courses. Core course registrations will get added by Workflow by default.
4. **Course Preference List File (optional input) (each row represents the preference list of 1 student)**

This file represents the course’s preference list, for it’s students. It is optional, and if it is not provided, it will be auto-generated by the software based on the ranking criteria provided in the ‘Course List’ file. Information about the ranking criteria is given in the ‘Important Terminology Used’ section.

* 1. Column 1 : Course Number followed by ‘$’, followed by whether it is an inside or outside department versions of the course (eg. CS5000$inside). More information on Inside department version of a course is given in the ‘Important Terminology Used’ section.
  2. Column 2 onwards : Comma separated list of students in the order of preference (highest to lowest)

1. **Max Credit Limit File (optional)**

Read ‘Important Terminology Used’ to know what Max Credit Limit is

* 1. Column 1 : Department and Year for students (eg. CS15) - This means that students whose roll number starts with ‘CS15’ will be considered for this rule
  2. Column 2 : Maximum Credit Limit for these students (eg. 63)

1. **Slot Configuration File (each row represents 1 slot)**

This file specifies the timings of each slot. It can change from semester to semester.

* 1. Column 1 : Name of the Slot (any slot name can be given. Hence you can create your own custom slot as well) (eg. A)
  2. Column 2 onwards : Comma separated list of timings for the slot.

An example of a row in this file will look like

A,MON=8:00-8:50,TUE=13:00-13:50,THU=11:00-11:50,FRI=10:00-10:50

1. **High Priority Students List File (optional)**

Read ‘Important Terminology Used’ to know what a High Priority student is

i) Column 1 : Name of the course (eg. HS3100)

ii) Column 2 onwards: Comma separated list of department and year whose students are considered high priority by this course. (eg. CS15)

An example of a row in this file will look like

HS2030,MM15,EP15,AE16,CH16

1. **Inside Department Specification File (each row represents 1 student)**

Read ‘Important Terminology Used’ to know what Inside department specification is

* 1. Column 1 : Name of the course (eg. HS3100)
  2. Column 2 onwards: Comma separated list of department and year whose students are considered high priority by this course. (eg. CS15)

Important Examples - these should be added to the file:

* + All OE department courses consider all NA students as inside department
  + All BT courses consider all BS and BE students as inside department.
  + Some EE courses might consider some EP students as inside department
  + All PH courses consider EP students as inside department

1. **Name of output Folder :**

This is the name of the folder where the output will be printed. Be careful to change the name every time you run the software, else the previous output will be overwritten.

1. **Algorithm to be used**

The software will also ask what algorithm should be used. If you are still uncomfortable with using the software just pick the ‘First Preference Allotment’ or ‘Iterative HR’ algorithms. If you comfortable with using the software, run the software once with each type of algorithm, and decide which output to use based on the statistics provided in the output folder

**4) Output Files given out by the Software**

1. **output.csv**

This file gives the final output of the Algorithm. It is represented in student-course pairs, where each pair corresponds to one allotment of a student to the course. Usually this is the output to be provided to workflow

i) Column 1 : Roll Number of the student (eg. CS13B031)

ii) Column 2 : Course Number of the course (eg. CS5001)

1. **perStudentAllottedCourses.csv**

This file gives the same information as the ‘output.csv’, but it is represented as the list of courses allotted for every student. This is useful to quickly scan the set of courses allotted to a particular student.

i) Column 1 : Roll Number of the student

ii) Column 2 onwards: comma separated list of courses allotted to that student

1. **perCourseAllottedStudents.csv**

This file gives the same information as the ‘output.csv’, but it is represented as the list of students allotted for every course. This is useful to quickly scan the set of students allotted to a particular course. Note that we separately mention the list of students for the inside and outside department course.

i) Column 1 : Course Number of the Course along with it’s inside/outside department information (eg. CS5001$inside)

ii) Column 2 onwards: comma separated list of students allotted to that course

1. **CoursePreferenceList.csv**

In case the course preference list file was auto generated by the software, it will be output here. In case you wish to re-run the allotment, use this file instead of re-generating the course preferences again because you may get a different course preference list (because some courses give a random ranking criteria, and ties are broken randomly)

i) Column 1 : Course Number of the Course along with it’s inside/outside department information (eg. CS5001$inside)

ii) Column 2 onwards: comma separated list of students from highest to lowest preference

1. **preferencesWithZeroCapacity.csv**

This file is not used as of now

1. **inputDataErrorLog.txt**

This file logs all the errors in the input file which were corrected by the software. This does not specify all the unrecoverable errors, like if a student’s GPA is missing, or if a file is not in the correct format. Those errors will be printed directly to the terminal shell or the GUI. These errors should be carefully analysed. Some examples of the types of errors provided by this file are as follows

i) If the student has provided a ‘Max Credits’ value which is higher than the limit set down by his department (or the default value of 60 in case his department has not specified any limit). Then his ‘Max Credits’ value is brought down to that limit.

ii) If the student has provided a course in his preference list which has a slot conflict with one of his core courses, and hence is pruned out of his preference list.

iii) If the student has provided a course in his preference list which has a credit value which exceeds the ‘Max Credits’ value (after taking into account the credits of his core courses) and hence is pruned out of his preference list.

iv) If a student has an empty preference list, or his preference list is emptied because of the pruning out in the above cases, and is not considered for the allotment.

1. **rejections.csv**

This file gives the number of students rejected by every course during the course of the allotment. This is simply a heuristic which approximately tells you which courses are popular (high demand courses). Higher the number of rejections, the more popular is the course. You need not use this file if you do not fully understand it’s purpose.

i) Column 1 : Course Number of the Course (eg CS5100$inside)

Course ID,Number of Rejections,

,Course Capacity,ii) Column 2 : Number of Rejections

iii) Column 3 : Number of First Preference Rejections (This specifies the number of rejections that happened when the course was at the beginning of the student’s preference list).

iv) Column 4 : Number of Rejections / Capacity

v) Column 5 :Number of First Preference Rejections / Capacity

1. **slotOrderingUsed.csv (optional)**

This file gives the slot ordering that was used for the SlotBased HR algorithms. It is not printed for the other algorithms. You need not use this file if you do not fully understand it’s purpose.

1. **perStudentStatistics.csv**

This file gives the statistics for every student. It mentions the ‘Effective Average Rank’ and ‘Credit Satisfaction Ratio’ for the student. Read ‘Important Terminology Used’ to know what these terms mean

i) Column 1 : Roll Number of the student

ii) Column 2 : Effective Average Rank of that student

iii) Column 3 : Credit Satisfaction Ratio for that student.

1. **aggregateStatistics.csv**

This file aggregates the statistics given in the perStudentStatistics.csv file. It computes the overall measures which can be used to compare the outputs of different algorithms, and decide which is better

It computes the average, variance and worst 10 percentile for the ‘Effective Average Rank’ and ‘Credit Satisfaction Ratio’ measures. We use variance because even if the mean is good, but the variance is very high, it means that the allotment has been unfair. Some students have got very good allotments, but some students have got very bad allotments.

**k) reasonsForNotAllottingPreferences.csv**

This file gives for every student preference that was not allotted, the detailed reason for not allotting that preference.

i) Column 1 : Roll Number of the student

ii) Column 2 : Course number of the preference not allotted

iii) Column 3 : Reason for not allotting that preference

**l) exchangeUnstablePairs.csv**

A pair of students who can exchange 2 of their courses, such that both of them will be better off, constitute an exchange unstable pair. We list here the set of all such exchange unstable pairs. IterativeHR algorithm can give exchange unstable pairs, but FirstPreferenceAllotment algorithm will guarantee to not give any such pairs.

**m) StudentEmails folder**

This folder contains 1 email message for every student. It consists of the list of allotted courses, and the list of courses not allotted, along with their reason. This reason will be a simplified version of the reason in the reasonsForNotAllottingPreferences.csv file

**5) Sample Input and Output Files**

**5a) Sample Input Files**

All the files use CSV (comma separated values) format.

**a) Student List File (each row represents 1 student)**

StudentRollNumber,GPA,MaxCredits

CS13B001,9.73,22

CS13B002,6.63,18

**b) Course List File (each row represents 1 course)**

CourseNumber,Department,TotalCapacity,OutsideDepartmentCapacity,RankingCriteria,Credits,Slots

CS0000,CS,0,0,1,1,F

CS1000,CS,1,0,3,9,D

CS2000,CS,1,0,1,12,D

CS3000,CS,1,0,1,12,A

**c) Student Preference List File (each row represents 1 Preference. I.e. 1 student-course pair)**

StudentRollNumber,CourseNumber,Colour,Type,PreferenceNumber

CS13B001,CS0000,NULL,CORE,NULL

CS13B001,CS1000,3,ELECTIVE,1

CS13B001,CS2000,1,ELECTIVE,2

CS13B001,CS3000,1,ELECTIVE,3

CS13B002,CS2000,0,ELECTIVE,1

CS13B002,CS3000,2,ELECTIVE,2

CS13B002,CS1000,0,ELECTIVE,3

**d) Course Preference List File (Optional - Can be generated from Ranking Criteria if you don’t want to use an existing one)**

Course Name, Preferences

CS0000$inside

CS0000$outside

CS1000$inside,CS13B001,CS13B002

CS1000$outside,CS13B001,CS13B002

CS2000$inside,CS13B001,CS13B002

CS2000$outside,CS13B001,CS13B002

CS3000$inside,CS13B001,CS13B002

CS3000$outside,CS13B001,CS13B002

**e) Max Credit Limit File (optional)**

Department&Year,Credit Limit

CS15,63

**f) Slot Configuration File (each row represents 1 slot)**

SlotName,Timings(comma seperated)

A,MON=8:00-8:50,TUE=13:00-13:50,THU=11:00-11:50,FRI=10:00-10:50

B,MON=9:00-9:50,TUE=8:00-8:50,WED=13:00-13:50,FRI=11:00-11:50

C,MON=10:00-10:50,TUE=9:00-9:50,WED=8:00-8:50,FRI=13:00-13:50

D,MON=11:00-11:50,TUE=10:00-10:50,WED=9:00-9:50,THU=13:00-13:50

E,TUE=11:00-11:50,WED=10:00-10:50,THU=8:00-8:50,FRI=16:50-17:40

F,TUE=16:50-17:40,WED=11:00-11:50,THU=9:00-9:50,FRI=8:00-8:50

G,MON=13:00-13:50,WED=16:50-17:40,THU=10:00-10:50,FRI=9:00-9:50

H,MON=14:00-15:15,WED=15:25-16:40

J,TUE=14:00-15:15,THU=15:25-16:40

K,TUE=15:25-16:40,WED=14:00-15:15

L,MON=15:25-16:40,THU=14:00-15:15

M,THU=16:50-18:05,FRI=14:00-15:15

P,MON=14:00-16:40

Q,TUE=14:00-16:40

R,WED=14:00-16:40

S,THU=14:00-16:40

T,FRI=14:00-16:40

A1,MON=8:00-8:50,THU=11:00-11:50,FRI=10:00-10:50,

B1,MON=9:00-9:50,TUE=8:00-8:50,FRI=11:00-11:50,

C1,MON=10:00-10:50,TUE=9:00-9:50,WED=8:00-8:50,

D1,MON=11:00-11:50,TUE=10:00-10:50,WED=9:00-9:50,

E1,TUE=11:00-11:50,WED=10:00-10:50,THU=8:00-8:50,

F1,WED=11:00-11:50,THU=9:00-9:50,FRI=8:00-8:50,

G1,THU=10:00-10:50,FRI=9:00-9:50

P1,MON=14:00-16:50

Q1,TUE=14:00-16:50

R1,WED=14:00-16:50

S1,THU=14:00-16:50

T1,FRI=14:00-16:50

A2,TUE=16:00-16:50,WED=13:00-13:50,FRI=15:00-15:50,

B2,WED=14:00-14:50,THU=13:00-13:50,FRI=16:00-16:50,

C2,MON=13:00-13:50,WED=15:00-15:50,THU=14:00-14:50,

D2,MON=14:00-14:50,WED=16:00-16:50,THU=15:00-15:50,

E2,MON=15:00-15:50,TUE=13:00-13:50,THU=16:00-16:50,

F2,MON=16:00-16:50,TUE=14:00-14:50,FRI=13:00-13:50,

G2,TUE=15:00-15:50,FRI=14:00-14:50

P2,MON=9:00-11:50

Q2,TUE=9:00-11:50

R2,WED=9:00-11:50

S2,THU=9:00-11:50

T2,FRI=9:00-11:50

**g) High Priority Students List File (optional)**

Course,Department&Year,

HS2030,MM15,EP15,AE16,CH16,CS16,EE16,MM16,NA16,EP16,BE16,BS16,PH16

HS3060,MM15,EP15,AE16,CH16,CS16,EE16,MM16,NA16,EP16,BE16,BS16,PH16

MA2020,MM15,EP15,AE16,CH16,CE16,

**h) Inside Department Specification File (each row represents 1 student)**

(REMEMBER : INCLUDE 'EP' IN 'PH' AND 'EE' COURSES. INCLUDE 'NA' IN 'OE' COURSES. INCLUDE 'BE','BS' IN 'BT' COURSES)

Course,Department&Year

BT1022,BS15,BS16,BE15,BE16

BT2010,BS15,BS16,BE15,BE16

EP2110,EP15,EP16

ID6030,EP15,EP16

PH2170,EP15,EP16

PH5110,EP15,EP16

PH6022,EP15,EP16

PH6022,EP15,EP16

OE3045,NA15,NA16

OE3190,NA15,NA16

**h) Name of output Folder**

Eg. SEAToutput

**i) Algorithm to be used**

Eg. 1

**5b) Output Files given out by the Software**

**a) output.csv**

Student Roll Number, Course ID

CS13B001,CS1000

CS13B001,CS3000

CS13B002,CS2000

**b) perStudentAllottedCourss.csv**

Student Roll Number, Electives Allotted

CS13B001,CS1000,CS3000

CS13B002,CS2000

**c) perCourseAllottedStudents.csv**

Course ID, Total Capacity, Allotted Capacity, Students Allotted

CS0000$inside,0,0

CS0000$outside,0,0

CS1000$inside,1,1,CS13B001

CS1000$outside,0,0

CS2000$inside,1,1,CS13B002

CS2000$outside,0,0

CS3000$inside,1,1,CS13B001

CS3000$outside,0,0

**d) CoursePreferenceList.csv**

Course Name, Preferences

CS0000$inside

CS0000$outside

CS1000$inside,CS13B001,CS13B002

CS1000$outside,CS13B001,CS13B002

CS2000$inside,CS13B001,CS13B002

CS2000$outside,CS13B001,CS13B002

CS3000$inside,CS13B001,CS13B002

CS3000$outside,CS13B001,CS13B002

**e) preferencesWithZeroCapacity.csv**

This file is not used as of now

**f) inputDataErrorLog.txt**

Encountered student with roll number NA15B005 exceeding his max credits. Student's max credits were reduced

Removing Preference since it exceeds credit limit. Student : AE15B002 - Course : CS3500$outside

**g) rejections.csv**

Course ID,Number of Rejections,Number of FirstPreference Rejections,Course Capacity,Number of Rejections / Capacity,Number of FirstPreference Rejections / Capacity

CS0000$inside,0,0,0,0.0,0.0

CS0000$outside,0,0,0,0.0,0.0

CS1000$inside,0,0,1,0.0,0.0

CS1000$outside,0,0,0,0.0,0.0

CS2000$inside,0,0,1,0.0,0.0

CS2000$outside,0,0,0,0.0,0.0

CS3000$inside,0,0,1,0.0,0.0

CS3000$outside,0,0,0,0.0,0.0

**h) slotOrderingUsed.csv (optional)**

D,A,A1,A2,B,B1,B2,C,C1,C2,D1,D2,E,E1,E2,F,F1,F2,G,G1,G2,H,J,K,L,M,P,P1,P2,Q,Q1,Q2,R,R1,R2,S,S1,S2,T,T1,T2,

**i) perStudentStatistics.csv**

Student Roll No,EffectiveAverageRank,creditSatisfactionRatio

CS13B001,0.75,1.0

CS13B002,1.0,0.67

**j) aggregateStatistics.csv**

EFFECTIVE RANK STATISTICS

Mean = 0.88

Standard Deviation = 0.13

Lowest 10Percentile = 1.0

CREDIT SATISFACTION RATIO STATISTICS

Mean = 0.83

Standard Deviation = 0.17

Lowestn10Percentile = 0.67

**k) reasonsForNotAllottingPreferences.csv**

Student Roll No,Course Number,Reason for Not allotting Preference

AE15B001,AS5212$outside,Already allotted to inside department version of the course

AE15B001,AS3050$inside,Course capacity is full.

AE15B002,AS5213$inside,Slot Clash with core course HS3050

**l) exchangeUnstablePairs.csv**

Student1 Roll No,Student1 Allotted Course,Student2 Roll No,Student2 Allotted Course

CS13B031, CS2000, CS13B000, CS5000

CS13B099, CS9000, CS13B090, CS8000

**6) Running the software from the terminal**

1. Navigate to the directory containing the ‘SEAT.jar’ file, using ‘Terminal’
2. Enter the command ‘java -jar SEAT.jar’
3. The locations of the all the input files mentioned in Section 3 will be asked (you can enter the name of the file directly if it is located in the same folder as the SEAT.jar file)
4. The algorithm will completed execution. Printed below is a sample input -> The parts in Bold are the ones printed by the software. The ones in regular fonts are your inputs

**Please enter the input csv file name where the slot timings are stored**

new\_slot\_config.csv

**Please enter the input csv file name where the list of students+CGPAs in the above format is stored**

studentList.csv

**Please enter the input csv file name where the list of courses+rankingCriteria+capacity in the above format is stored**

courseListModified.csv

**Please enter the input csv file name where the preference list in the above format are stored**

studentPreferenceList.csv

**Course Preference List Generation. Please enter 1 to create New Course Preference Lists, 2 to read from existing Course Preference List from a file:**

1

**Please enter the location of file where you want to save the freshly generated course preferences:**

coursePreferenceList.csv

**Which algorithm to run?**

**1. Iterative HR**

**2. First Preference Allotment**

**3. Slotwise HR (with Heuristic 1)**

**4. Slotwise HR (with Heuristic 2)**

1

**Please enter output folder to print to**

SEAToutput

**Please enter the input csv file name where the list of courses that consider other department students as inside department. If you do not require this functionality, just enter the name of a blank csv file.**

insideDepartmentSpecificationModified.csv

**Please enter the input csv file name where the list of courses that consider certain students as high priority students. If you do not require this functionality, just enter the name of a blank csv file.**

highPriorityStudentsModified.csv

**Please enter the input csv file name where the the deparment&year wise limit on the max credits for students**

maxCreditLimits.csv

**Populating data.......**

**Sanitizing Input......**

**Input Data has errors. Error log printed to the file 'InputDataErrorLog'**

**Enter 1 if you wish to continue with the errors in the preference lists pruned out**

1

**Running the Iterative HR algorithm......**

**Sanitizing Output ....**

**Execution over.......**

**7) Running the software from the GUI (TO DO ONCE GUI IS COMPLETED)**

**8 ) Modifications to Workflow data to fit it to the Software**

These are some manual checks (could be automated by small scripts, but they are not done by the main SEAT software because of their varying nature) that must be done on the Input data before feeding it to the SEAT software.

1. **Use only Registered courses**

Try to use only those courses for which students have given preferences or registered for as core courses. The Slotbook will contain several other extra courses and we should not waste time trying to fix the discrepancies and issues for courses that no one has given preferences for.

1. **Deal with duplicate course (also use only registered courses)**

Data provided in the Slotbook, typically has duplicate entries for courses that run in multiple batches or classrooms, and this usually occurs for core courses. These duplicates need to merged into a single entry. There could be ambiguity regarding the slots for the course, if the different entries have different slots. The could also be ambiguity regarding the capacity because if say 3 entries of the course have a capacity of 15 each, then the single entry might have capacity of 15 or 45. These doubts must be clarified with the department that offers the course.

1. **Block only specific slot for courses that run in multiple slots wherever possible**

Some courses run in multiple slots and multiple batches, but each batch runs in only a few of those slots. For example, the course CS1100 could run in the A,B and C slots. But only the first 10 AE students take the course in the A slot, the next 10 students take it in the B slot and the next 10 students in the C slots. In this case, we should not block all three slots for all the students taking the course. Instead we should create 3 versions of the course - ‘CS1100slotA’, ‘CS1100slotB’ and CS1100slotC’, and have a separate entry in the course list for each of these courses. We also modify the student preference list to use these 3 versions instead of CS1100. To decide which version to give a particular student, we look at the registration data provided by workflow. There will be a column for ‘Slot’, which is otherwise ignored by us, but here we will use this slot to decide which version of the course to add to the student’s preference list. In the above example, the ‘Slot’ column in the registration data provided by workflow will have the entry ‘A’ for the course CS1100 for the first 10 AE students. Hence, we need to modify the course number for all AE students to ‘CS1100slotA’. We similarly modify the course number of the next 10 students to ‘CS1100slotB’, and to CS1100slotC’ for the last 10 students.

1. **Check for missing preference numbers or repeated preference numbers**

The registration data provided by workflow could have gaps in the preference list, or repeated preference numbers. This missing data needs to be clarified from Workflow.

1. **Check for unusual slots**

Some courses run in unusual slots like XX,YY,TT, etc. We need to confirm exactly what time these courses run at.

1. **Check that students have only registered for courses for which btech/dd = 1**

The Slotbook has a column called ‘Btech/DD’, whose entry will be 1 for all courses which are open for SEAT allocation. We need to ensure that all elective course preferences given by students have the corresponding ‘Btech/DD’ entry set to 1 in the Slotbook.

1. **Max credit default limit**

Apart from the department specific max credit limit provided in the maxCreditLimit input file, there is a default credit limit which has to be modified in the Constants.java file. Ensure that this is set to the correct value.

1. **Courses with zero capacity, but students who give preferences for it.**

There are sometimes courses with 0 capacity, but preferences for it. Or there could be a 0 outside department capacity, but with outside department students who register for it. Sometimes, a course could belong to a department like AM, which does not have any Btech/DD students, and the outside department capacity of that course is 0. This means that nobody will be able to register for that course. All such cases should be clarified with the department offering the course.

1. **Courses with high rejections**

Any course with more than 10 rejections needs to be looked into more carefully. There could possibly be a mistake in specifying the capacity. It needs to be dealt with on a case by case basis.