

---

# CS246: Database Management Systems Lab

Lab # 05 (1 Questions, 50 Points)

Timings: 14:00 to 17:00 Hours    Pages: 4

**IIT Guwahati**

04 Feb 2020 (Tue)

---

## Question 1: (50 points)

Implement the following problem in either C or C++ Programming Language. Use of any other programming language is not allowed and will lead to awarding 0 marks. Use of standard library functions to perform file operations lead to awarding 0 marks.

**Input Data** You are given three input data files containing the following:

**students03.csv** Each line contains information about one student as described below:

- Student name
- Student roll number

*Every row of students file contains 41 bytes*

**courses03.csv** Each line contains information about one course. The following attributes are given for each course

- Semester number in which the course is offered. Semester number is between 1 and 8 (both inclusive)
- Course number
- Course name
- Number of lecture hours per week
- Number of tutorial hours per week
- Number of practical hours per week
- Total number of credits

*Every row of courses file contains 67 bytes*

**grades03.csv** Every line contains the following information

- Student roll number
- Course number
- Grade the student obtained in this course

*Every row of grades file contains 20 bytes*

**Disclaimer** The grades data is randomly generated and has no bearing with the actual data.

**Constraints** Following constraints must be followed

- You must allocate 4096 bytes of memory for holding **students** records. Allocate memory dynamically of this size. You must **free** the memory as and when it is not needed.

- You must allocate 8192 bytes of memory for holding **courses** records. Allocate memory dynamically of this size. You must **free** the memory as and when it is not needed.
- You must allocate 20 bytes of memory for holding **grades** records. Allocate memory dynamically of this size. You must **free** the memory as and when it is not needed.
- You can use float array which can hold SPI and CPI.
- Use of excessive memory is discouraged.

**Problem Statement** Your task is to:

**Read** These three files; You must use **open** system call with arguments *read only* mode.

**Transcripts** For each student produce the *transcript* (a student's view of the data)

**Write** Save the above transcript view into respective **roll\_number.txt** file. Every transcript file you create must have the following permissions:

- You should use **open** system call in write only mode.
- If the file already exists, you must over-write the contents of file.
- You should always append data to the file.
- Perform synchronous writes.
- You should instruct the kernel that write be made with **Synchronized I/O *data integrity*** mode
- You should provide the following permissions at the time of opening the file.

**Owner** Has Read and write permission but no executable permission.

**Group** Has Read and executable permissions but no write permission.

**Other** Has no read, write and executable permission.

**Transcript format** Your program output should produce the following transcript format:

Name:

Roll Number:

Semester I

Course Number	Course Name	C	Grade
PH101	Physics - 1	6	CC
PH110	Physics Lab	3	CD

..

Semester II

Course Number	Course Name	C	Grade
PH101	Physics - 2	6	CD
MA102	Mathematics-2	6	BB

..

I      II      III ...      VIII

SPI	7.59	5.86	..	..
CPI	7.59	6.70	..	..

**Accounting** You must output number of disk reads performed on

- students03.csv file
- courses03.csv file
- grades03.csv file

Note moving file pointer do not account for disk read.

**Accounting** You must output number of disk writes performed to produce the transcript for each student.

**Accounting** You must output number of lseek performed to on grades03.csv file.

**Observations and Hints** Following are important observations and hints you may want to use

**Observations 1** students03.csv file is sorted by roll number. You can use this information in preparing transcripts one student at a time.

**Observations 2** students03.csv file size is 6478 bytes which is more than 4096 bytes. You must read this file two times (using loop structure).

**Observations 3** courses03.csv file is sorted according to semester number.

**Observations 4** grades03.csv is sorted by semester number, course number followed by roll number.

**Hint 1** You may want to obtain transcripts one student at a time given the memory constraints on the students. For each student (as per roll number) obtain his/her complete transcript. You can then move on to computing next roll number's transcript.

**Hint 2** Every student record in the grades03.csv file is located at fixed bytes. Compute this fixed byte location and then jump to that exact position using lseek.

**Instructions** Adhere to the following

**File naming** Prepend C/C++ program file names with your roll number. Adhere to the input and output file naming convention as given in the problem description.

**Independent efforts** You should make an honest and independent effort in obtaining the solution to the above problem. You are also encouraged to bring one data structures and algorithms text book and one programming language text book of your choice.

**Discussions** with fellow students are not allowed.

**Internet** Use of internet during lab hours is not allowed.

**Mobile phones** Use of mobile phones in the lab hours is not allowed.

**Evaluation** At the end of 17:00 hours, TAs will come and evaluate your program. Leave the lab once your evaluation is completed.

**Marking Scheme** The evaluation criteria is as follows:

**3 Marks** For reading three input files (using read system call)

**12 Marks** For producing for each student semester-wise grades

- (5 marks) Locating grade of this student for the next course in the `courses03.csv` file
- (3 marks) Opening the student transcript file (`rollnumber.txt`) for writing with all specified options
- (4 marks) Obtaining *all* students transcripts. Including reading the students file *completely*.

**3 Marks** For freeing the all allocated memory

- (1 marks) For freeing students memory
- (1 marks) For freeing courses memory
- (1 mark) For freeing grades memory

**2 Marks** For closing all open file descriptors

**2 Marks** For computing SPI

**2 Marks** For computing CPI

**5 Marks** Allowing Synchronized I/O data integrity

**3 Marks** For read accounting

**2 Marks** For write accounting

**3 Marks** For lseek accounting

**5 Marks** For producing all students transcripts

**5 Marks** Correct output

**3 Marks** For adhering to memory constraints