

<b>FILE STRUCTURES LABORATORY WITH MINI PROJECT</b> <b>[As per Choice Based Credit System (CBCS) scheme]</b> <b>(Effective from the academic year 2016 -2017)</b> <b>SEMESTER – VI</b>			
Subject Code	15ISL68	IA Marks	20
Number of Lecture Hours/Week	01I + 02P	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03
<b>CREDITS – 02</b>			
<b>Course objectives:</b> This course will enable students to			
<ul style="list-style-type: none"> <li>• Apply the concepts of Unix IPC to implement a given function.</li> <li>• Measure the performance of different file structures</li> <li>• Write a program to manage operations on given file system.</li> <li>• Demonstrate hashing and indexing techniques</li> </ul>			
<b>Description (If any):</b>			
Design, develop, and implement the following programs			
<b>Lab Experiments:</b>			
<b>PART A</b>			
<ol style="list-style-type: none"> <li>1. Write a program to read series of names, one per line, from standard input and write these names spelled in reverse order to the standard output using I/O redirection and pipes. Repeat the exercise using an input file specified by the user instead of the standard input and using an output file specified by the user instead of the standard output.</li> <li>2. Write a program to read and write student objects with fixed-length records and the fields delimited by " ". Implement pack ( ), unpack ( ), modify ( ) and search ( ) methods.</li> <li>3. Write a program to read and write student objects with Variable - Length records using any suitable record structure. Implement pack ( ), unpack ( ), modify ( ) and search ( ) methods.</li> <li>4. Write a program to write student objects with Variable - Length records using any suitable record structure and to read from this file a student record using RRN.</li> <li>5. Write a program to implement simple index on primary key for a file of student objects. Implement add ( ), search ( ), delete ( ) using the index.</li> <li>6. Write a program to implement index on secondary key, the name, for a file of student objects. Implement add ( ), search ( ), delete ( ) using the secondary index.</li> <li>7. Write a program to read two lists of names and then match the names in the two lists using Consequential Match based on a single loop. Output the names common to both the lists.</li> <li>8. Write a program to read k Lists of names and merge them using k-way merge algorithm with k = 8.</li> </ol>			
<b>Part B --- Mini project:</b>			
Student should develop mini project on the topics mentioned below or similar applications <b>Document processing, transaction management, indexing and hashing, buffer management, configuration management. Not limited to these.</b>			
<b>Course outcomes:</b> The students should be able to:			

- Implement operations related to files
- Apply the concepts of file system to produce the given application.
- Evaluate performance of various file systems on given parameters.

**Conduction of Practical Examination:**

1. All laboratory experiments from part A are to be included for practical examination.
2. Mini project has to be evaluated for 30 Marks as per 6(b).
3. Report should be prepared in a standard format prescribed for project work.
4. Students are allowed to pick one experiment from the lot.
5. Strictly follow the instructions as printed on the cover page of answer script.
6. Marks distribution:
  - a) Part A: Procedure + Conduction + Viva: 10 + 35 + 5 = 50 Marks
  - b) Part B: Demonstration + Report + Viva voce = 15 + 10 + 05 = 30 Marks
7. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.