Computer Science Courses for Undergraduate Programme of study with Computer Science discipline as one of the two Core Disciplines
(For e.g. courses for B.A. Programmes with Computer Science as Major discipline)

Computer Science Major

DISCIPLINE SPECIFIC CORE COURSE (DSC04): Operating Systems

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSC04: Operating Systems	**4**	**3**	**0**	**1**	Pass in Class XII	DSC 01 Programming using Python/ A course in C/C++/Python at plus 2 level

Learning Objectives

This course introduces the students to Operating Systems and its importance in computer systems. The focus is to explain the common services provided by an operating system like process management, memory (primary, secondary & virtual) management, I/O management, file management. The course talks about the various functional components of the operating and their design.

Learning outcomes

On successful completion of the course, students will be able to

- gain knowledge of different concepts of the operating System and its components.
- learn about shell scripts and would be able to use the system in an efficient manner.

SYLLABUS OF DSC04

Unit 1 (4 hours)

Introduction: Operating Systems (OS) definition and its purpose, Multiprogrammed and Time Sharing Systems, OS Structure, OS Operations: Dual and Multi-mode, OS as resource manager.

Unit 2 (10 hours)

Operating System Structures: OS Services, System Calls: Process Control, File Management, Device Management, and Information Maintenance, Inter-process Communication, and Protection, System programs, OS structure- Simple, Layered, Microkernel, and Modular.

Unit 3 (9 hours)

Process Management: Process Concept, States. Process Control Block, Context Switch, Process scheduling, Schedulers, Overview of threads and Scheduling Algorithms: First Come First Served, Shortest-Job-First, Priority & Round-Robin.

Unit 4 (9 hours)

Memory Management: Physical and Logical address space, Swapping Contiguous memory allocation strategies - fixed and variable partitions, Segmentation, Paging, virtual memory: Demand Paging.

Unit 5 (8 hours)

File and Input / Output Device Management: File Concepts, File Attributes, File Access Methods, Directory Structure: Single-Level, Two-Level, Tree-Structured, and Acyclic-Graph Directories, Magnetic Disks, Solid-State Disks, Magnetic Tapes.

Unit 6 (5 hours)

Shell Scripting: Shell variables, parameter passing conditional statements, iterative statements, writing and executing shell scripts, utility programs (cut, paste, grep, echo, pipe, filter etc.)

Essential/recommended readings

1. Galvin, S. P. B., Gagne, G., *Operating System Concepts*, 9th edition, John Wiley Publications, 2016.
2. G. Nutt, Operating Systems, Pearson, 2009
3. Das, S., *Unix: Concepts and Applications*, 4th edition, TMH, 2009.

Additional References

1. Dhamdhere, D. M., *Operating Systems: A Concept-based Approach,* 2nd edition, Tata McGraw-Hill Education, 2017.
2. Kernighan, B. W., Pike, R., *The Unix Programming Environment*, Englewood Cliffs, NJ: Prentice-Hall, 1984.
3. Stallings, W., *Operating Systems: Internals and Design Principles*, 9th edition, Pearson Education, 2018.
4. Tanenbaum, A. S., *Modern Operating Systems.* 3rd edition, Pearson Education, 2007.

Suggested Practical List (If any): (30 Hours)

Practical exercises such as

1. Execute various LINUX commands for:

    i. Information Maintenance: wc, clear, cal, who, date, pwd

    iv. File Management: cat, cp, rm, mv, cmp, comm, diff, find, grep

    v. Directory Management : cd, mkdir, rmdir, ls

2. Execute various LINUX commands for:

    iv. Process Control: fork, getpid, ps

    v. Communication: Input-output redirection, Pipe

    vi. Protection Management: chmod, chown, chgrp

3. Write a program(using fork() and/or exec() commands) where parent and child execute:

    iv. same program, same code.

    v. same program, different code.

    vi. before terminating, the parent waits for the child to finish its task.

4. Write a program to calculate sum of n numbers using Pthreads.

5. Write a program to generate a Fibonacci Series of numbers using Pthreads.

6. Write a program to implement best-fit and worst-fit allocation strategies

7. Write a program to copy files using system calls and using pthreads and compare timings.

8. Write a program to implement FCFS scheduling algorithm.

9. Write a program to implement SJF scheduling algorithm.

10. Write a program to implement non-preemptive priority based scheduling algorithm.

DISCIPLINE SPECIFIC CORE COURSE (A4): Data Mining-II

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
A4: Data Mining - II	4	3	0	1	Pass in Class XII	DSC01 Programming using Python, / GE1b

						Programming with Python / A1 Programming Fundamentals using Python ,Data Mining-I

Learning Objectives

The course introduces the students to the supervised and unsupervised learning techniques. Students will learn about the importance of ensemble methods, cluster analysis, anomaly detection and their applicability in mining patterns in real applications. At the end students will be exposed to two advanced topics: text mining and time-series mining. Students will use the learned topics in solving real applications using open-source software.

Learning outcomes

On successful completion of the course, students will be able to:

- Differentiate between partition-based, density-based and hierarchical clustering
- Build ensemble models to improve predictive performance of the classifier
- Identify anomalies and outliers using supervised and unsupervised techniques
- Analyze time-series data and extract patterns from the stamped data
- Mine textual data and do topic modeling

SYLLABUS OF A4

Unit 1 (9 hours)

Clustering: Partitioning Methods, Hierarchical Methods, Density-Based Methods, Comparison of different methods

Unit 2 (8 hours)

Ensemble Methods: Need of ensemble, Random Forests, Bagging and Boosting

Unit 3 (10 hours)

Anomaly Detection: Outliers and Outlier Analysis, Outlier Detection Methods, Statistical Approaches, Proximity-based and density-based outlier detection, Clustering-based approaches

Unit 4 (8 hours)

Mining Text Data: Document Preparation and Similarity, Clustering Methods for Text, Topic Modeling

Unit 5 (10 hours)

Stream Mining: Time series basics, Date Ranges, Frequencies, and Shifting, Resampling and moving windows functions, Decay function, Clustering stamped data: STREAM and CluStream

Essential/recommended readings

1. Tan P.N., Steinbach M, Karpatne A. and Kumar V. *Introduction to Data Mining,* 2nd edition, Pearson, 2019.
2. Zaki M. J. and Meira J. Jr. *Data Mining and Machine Learning: Fundamental Concepts and Algorithms*, 2nd edition, Cambridge University Press, 2020.
3. Aggarwal C. C. *Data Mining: The Textbook*, Springer, 2015.

Additional References

1. Han J. Kamber M. and Pei J. *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, 2011.
2. Dunham M. *Data Mining: Introductory and Advanced Topics*, Pearson, 2006.

Suggested Practicals List (If any): (30 Hours)

Practical exercise such as

1. Apply Partitioning Methods, Hierarchical Methods, Density-Based Methods for clustering on a data set and compare the performance of the obtained results using different metrics
2. Create an ensemble using Random Forest and show the impact of bagging and boosting on the performance
3. Apply different outlier-detection methods on a noisy dataset and compare their effectiveness in terms of outliers reported
4. Compute similarity between two documents after required document preparation
5. Considering a time-stamped data (sales data/weather data), compare the aggregate values visually using different moving windows function
6. Write a program to find the latent topics in a document using any topic modeling method and display top 5 terms that contribute to each topic along with their strength. Also, visualize the distribution of terms contributing to the topics.

Project: Students should be promoted to take up one project covering at least one unit of the syllabus on any UCI/kaggle/data.gov.in or a dataset verified by the teacher. This will allow the students to have a practical knowledge of how to apply the various skills learnt in the subject for a single problem/project.