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)

DISCIPLINE SPECIFIC CORE COURSE (DSC-3): Computer System Architecture

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		
DSC03: Computer System Architecture	4	3	0	1	Passed 12th class with Mathematics	NIL

Learning Objectives

This course introduces students to the fundamental concepts of digital computer organization, design, and architecture. It aims to develop a basic understanding of the building blocks of a computer system and highlights how these blocks are organized together to architect a digital computer system.

Learning outcomes

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

- Design combinatorial circuits using basic building blocks. Simplify these circuits using Boolean algebra and Karnaugh maps. Differentiate between combinational circuits and sequential circuits.
- Represent data in binary form, convert numeric data between different number systems, and perform arithmetic operations in binary.
- Determine various stages of the instruction cycle and describe interrupts and their handling.
- Explain how the CPU communicates with memory and I/O devices.
- Simulate the design of a basic computer using a software tool.

SYLLABUS OF DSC-3

Unit 1 (9 hours)

Digital Logic Circuits: Digital Logic Gates, Flip flops and their characteristic table, Logic circuit simplification using Boolean algebra and Karnaugh map, Don't care conditions, Combinational circuits, Introduction to Sequential Circuits

Unit 2 (7 hours)

Digital Components: Decoders, Encoders, Multiplexers, Binary Adder, Binary Adder Subtractor, Binary Incrementor, Registers, and Memory Units

Unit 3 (13 hours)

Data Representation: Binary representation of both numeric and alphanumeric data, representation of numeric data in different number systems, (Binary, Octal, Decimal and Hexadecimal), conversion from one number system to another, complements, representation of signed and unsigned numbers, addition and subtraction of signed and unsigned numbers and overflow detection.

Unit 4 (9 hours)

Basic Computer Organization and Design: Stored program organization, Computer registers, Instruction set and their completeness, Instruction cycle, Memory reference instructions, Register reference instructions, Input- Output reference instructions, Interrupt cycle, Addressing modes.

Unit 5 (7 hours)

Input-Output Organization: I/O interface, I/O vs. Memory Bus, Isolated I/O, Memory Mapped I/O, Direct Memory Access.

Essential/recommended readings

1. M. Morris Mano, *Computer System Architecture*, 3rd edition, Pearson Education, 2017.
2. Linda Null, Julia Lobur, *Essentials of Computer Organization and Architecture*, 5th Edition, 2019.

Additional References

2. D. Comer, *Essentials of Computer Architecture,* 2nd edition, CRC Press, 2017.

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

Practical exercises such as

(Use Simulator – CPU Sim 3.6.9 or any higher version for the implementation)

1. Create a machine based on the following architecture:

Registers

IR	DR	AC	AR	PC	I	E
16 bits	16 bits	16 bits	12 bits	12 bits	1 bit	1 bit

Memory 4096 words	

16 bits per word	Instruction format 15 12 11 0 Opcode | Address

Basic Computer Instructions

Memory Reference			Register Reference	
Symbol	Hex		Symbol	Hex
AND	0xxx	Direct Addressing	CLA	7800
ADD	1xxx		CLE	7400
LDA	2xxx		CMA	7200
STA	3xxx		CME	7100
			HLT	7001

Refer to Chapter-5 for a description of the instructions.

Design the register set, the memory, and the instruction set. Use this machine for the assignments in this section.

1. Implement fetch sequence
2. Write an assembly program to simulate the addition of two numbers when one is stored in memory and another is entered by the user.
3. Write an assembly program to simulate addition of two numbers when both numbers are taken as inputs from user.
4. Write an assembly program to simulate subtraction of two numbers when one number is stored in memory and another is entered by the user.

5. Write an assembly program to simulate subtraction of two numbers when both numbers are taken as inputs from user

6. Write an assembly program to simulate the following logical operations on two user-entered numbers.

i.AND

ii.OR

iii.NOT

7. Write an assembly language program to simulate the machine for following register reference instructions and determine the contents of AC, E, PC, AR and IR registers in decimal after the execution:

i. CLE

ii. CLA

iii. CMA

iv. CME

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.

DISCIPLINE SPECIFIC CORE COURSE : Data Mining-I

-

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		
Data Mining - I	4	3	0	1	Passed 12th class with Mathematics	Programming using Python

Learning Objectives

This course aims to introduce data mining techniques and their application on real-life datasets. The students will learn to pre-process the dataset and make it ready for application

of data mining techniques. The course will focus on three main techniques of data mining i.e. Classification, Clustering and Association Rule Mining. Different algorithms for these techniques will be discussed along with appropriate evaluation metrics to judge the performance of the results delivered.

Learning outcomes

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

- Pre-process the data for subsequent data mining tasks
- Apply a suitable classification algorithm to train the classifier and evaluate its performance.
- Apply appropriate clustering algorithm to cluster the data and evaluate clustering quality
- Use association rule mining algorithms and generate frequent item-sets and association rules

SYLLABUS

Unit 1 (8 hours)

Introduction to Data Mining: Motivation and Challenges for data mining, Types of data mining tasks, Applications of data mining, Data measurements, Data quality, Supervised vs. unsupervised techniques

Unit 2 (9 hours)

Data Pre-Processing: Data aggregation, sampling, dimensionality reduction, feature subset selection, feature creation, variable transformation.

Unit 3 (11 hours)

Cluster Analysis: Basic concepts of clustering, measure of similarity, types of clusters and clustering methods, K-means algorithm, measures for cluster validation, determine optimal number of clusters

Unit 4 (8 hours)

Association Rule Mining: Transaction data-set, frequent itemset, support measure, rule generation, confidence of association rule, Apriori algorithm, Apriori principle

Unit 5 (9 hours)

Classification: Naive Bayes classifier, Nearest Neighbour classifier, decision tree, overfitting, confusion matrix, evaluation metrics and model evaluation.

Essential/recommended readings

1. Tan P.N., Steinbach M, Karpatne A. and Kumar V. *Introduction to Data Mining,* 2nd edition, Pearson, 2021.
2. Han J., Kamber M. and Pei J. *Data Mining: Concepts and Techniques*, 3rd edition, 2011, Morgan Kaufmann Publishers.
3. Zaki M. J. and Meira J. Jr. *Data Mining and Machine Learning: Fundamental Concepts and Algorithms*, 2nd edition, Cambridge University Press, 2020.

Additional References

1. Aggarwal C. C. *Data Mining: The Textbook*, Springer, 2015.

2. Dunham M. *Data Mining: Introductory and Advanced Topics,* 1st edition, Pearson Education India, 2006.

Recommended Datasets for :

Classification: Abalone, Artificial Characters, Breast Cancer Wisconsin (Diagnostic)

Clustering: Grammatical Facial Expressions, HTRU2, Perfume data

Association Rule Mining: MovieLens, Titanics

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

Practical exercise such as

1. Apply data cleaning techniques on any dataset (e,g, wine dataset). Techniques may include handling missing values, outliers, inconsistent values. A set of validation rules can be prepared based on the dataset and validations can be performed.
2. Apply data pre-processing techniques such as standardization/normalization, transformation, aggregation, discretization/binarization, sampling etc. on any dataset
3. Run Apriori algorithm to find frequent itemsets and association rules on 2 real datasets and use appropriate evaluation measures to compute correctness of obtained patterns
   a) Use minimum support as 50% and minimum confidence as 75%
   b) Use minimum support as 60% and minimum confidence as 60 %
4. Use Naive bayes, K-nearest, and Decision tree classification algorithms and build classifiers on any two datasets. Divide the data set into training and test set. Compare the accuracy of the different classifiers under the following situations:
   I. a) Training set = 75% Test set = 25% b) Training set = 66.6% (2/3rd of total), Test set = 33.3%
   II. Training set is chosen by i) hold out method ii) Random subsampling iii) Cross-Validation. Compare the accuracy of the classifiers obtained.

   Data is scaled to standard format.
5. Use Simple K-means algorithm for clustering on any dataset. Compare the performance of clusters by changing the parameters involved in the algorithm. Plot MSE computed after each iteration using a line plot for any set of parameters.

Project: Students should be promoted to take up one project on any UCI/kaggle/data.gov.in or a dataset verified by the teacher. Preprocessing steps and at least one data mining technique should be shown on the selected dataset. 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.

Note: Examination scheme and mode shall be as prescribed by the Examination Branch, University of Delhi, from time to time.