UNIVERSITY OF DELHI

CNC-II/093/1/Misc./2025/6
Dated: 10.06.2025

NOTIFICATION

Sub: Amendment to Ordinance V

The following modifications/ amendments in the syllabi of Department of Computer Science under Faculty of Mathematical Science based on UGCF 2022 are notified herewith for the information of all concerned:

1. Renaming of Courses in computer science programs under UGCF (for the academic session 2025-2026 onwards)

Existing	**Replace with**
Programming Using Python	Object Oriented Programming using Python. (As per Annexure-1)
Python Programming for Data Handling	Object Oriented Programming using Python
Data Exploration and Visualization	Data Analysis and Visualization using Python
Data Mining for Knowledge Discovery (DMKD) in Sem V	Data Mining-I
Applied Network Analytics	Social Network Analytics
Foundations of Computer Graphics	Computer Graphics
Data Interpretation and Visualization using Python	Data Analysis and Visualization using Python

2. DSEs to be added/deleted in the pool of Semester VII, VIII in UG Curriculum (Admissions 2022 and 2023)

Semester	Add/Delete in the pool of DSEs	Remarks
VII	Research Methodology (to be added)	These courses have already been approved in the AC/EC meetings.
	Machine learning (to be added)	

	Deep learning (to be added)	
	Reinforcement Learning (to be shifted to Semester VIII)	
	Computer Graphics (to be added)	
	Social Network Analytics (to be added)	
VIII	Reinforcement Learning (to be added)	
	Ethical Hacking (to be added)	

3. Revised syllabus of "Object Oriented Programming Using Python" as per **Annexure I**.
4. Replace "Data Structures using Python" with "Data Structures using C++" everywhere.
5. W.e.f. from admission year 2024, in BSc(H) Computer Science
   a. Replace Computer Graphics with Artificial Intelligence in Semester III
   b. Replace Theory of Computing with Machine Learning in Semester V.
   c. Replace Artificial Intelligence with Theory of Computing in Semester -VI
   d. Replace Machine Learning with Computer Graphics in Semester VI.
   e. Replace Cloud Computing with Deep Learning in Semester VI.
   f. Replace Compiler Design with Cloud Computing in Semester VII.

6. W.e.f. from admission year 2024, in BSc(P)/BSc(Math. Sc.)/BA(P) with Computer Science
   a. Replace Design and Analysis of Algorithms (DAA) with Software Engineering in Semester VII.
   b. Replace Computer system Architecture with Design and Analysis of Algorithms in Semester III
7. W.e.f. admission year 2024, List of DSEs

Odd		Even	
Sem III		Sem IV	
Object Oriented Programming using Python, Data Mining, Artificial Intelligence Android Programming using		Data Analysis and Visualization using Python, Artificial Intelligence, Combinatorial Optimization, Introduction to Web	

Java, Cyber Security		Programming Graph Theory Network Security	
Sem V		Sem VI	
Algorithms and Advanced Data Structures Machine Learning, Data Mining, Data Privacy Web Programming Unix Network Programming Web Design and Development Quantum Computing		Theory of Computation , Deep Learning Computer Graphics Ethical Hacking, Social Network Analytics, Research Methodology Cyber Forensics	
Sem VII		Sem VIII	
Digital Image Processing Advanced Algorithms Cyber Forensics Research Methodology Machine learning Deep learning Computer Graphics Social Network Analytics Compiler Design		Information and Image Retrieval Natural Language Processing BlockChain and its application Cloud Computing Reinforcement Learning Ethical Hacking Deep learning Computer Graphics	

8. W.e.f. admission year 2024, list of GEs (no change in first year)

Odd		Even	
Sem I		Sem II	
Programming Using Python		Data Analysis and Visualization using Python	
Programming Using C++		Data Analysis and Visualization using Spreadsheet	
		Computer System Architecture	
Sem III		Sem IV	
Database Management Systems		Data structures using C++	
JAVA Programming		Introduction to Web Programming	
		Software Engineering	
Sem V		Sem VI	
Operating Systems		Computer Networks	
Advanced Web Programming		Advanced Web Programming	
Java Based Web App Development		Artificial Intelligence	
		Data Privacy	

9. Syllabus of Data Mining as per **Annexure II**.

23/6/23

REGISTRAR

Annexure I

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSC 01 **Object Oriented Programming using Python**	**4**	**3**	**0**	**1**	Pass in Class XII	

Course Objective

This course is designed as the first course that introduces object oriented programming concepts using Python to Computer Science students. The course focuses on the development of Python programming to solve problems of different domains using object- oriented programming paradigm.

Course Learning Outcomes

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

1. Understand the basics of programming language
2. Develop, document, and debug modular Python programs.
3. Apply suitable programming constructs and built-in data structures to solve a problem.
4. Use and apply various data objects in Python.
5. Use classes and objects in application programs and handle files.
6. apply OOPs concepts such as encapsulation, inheritance and polymorphism in writing programs.

Syllabus

Unit 1 **(4 hours)**

Introduction to Programming: Problem solving strategies; Structure of a Python program; Syntax and semantics; Executing simple programs in Python.

Unit 2 **(10 hours)**

Creating Python Programs: Identifiers and keywords; Literals, numbers, and strings; Operators; Expressions; Input/output statements; Defining functions; Control structures (conditional statements, loop control statements, break, continue and pass, exit function), default arguments.

Unit 3 **(15 hours)**

Built-in data structures: Mutable and immutable objects; Strings, built-in functions for string, string traversal, string operators and operations; Lists creation, traversal, slicing and splitting operations, passing list to a function; Tuples, sets, dictionaries and their operations.

Unit 4 **(10 hours)**

Object Oriented Programming: abstraction, encapsulation, objects, classes, methods, constructors, inheritance, polymorphism, static and dynamic binding, overloading, abstract classes, interfaces and packages.

Unit 5 **(6 hours)**

File and exception handling: File handling through libraries; Errors and exception handling.

References

1. Allen B. Downey,**Think Python: How to Think Like a Computer Scientist,** O'Reilly Media, 2024.
2. J.V. Guttag, **Introduction to Computation and Programming Using Python: With Application to Understanding Data**, MIT Press, 2016.
3. Robert Sedgewick , Kevin Wayne, Robert Dondero, **Introduction to Programming in Python: An Interdisciplinary Approach,** Addison-Wesley Professional, 2015
4. Tony Gaddis, **Starting Out with Python**, Pearson, 2021.

Additional References

(i) Brown, Martin C. *Python: The Complete Reference*, 2nd edition, McGraw Hill Education, 2018.

Suggested Practical List

1. WAP to find the roots of a quadratic equation
2. WAP to accept a number 'n' and
    a. Check if 'n' is prime
    b. Generate all prime numbers till 'n'
    c. Generate first 'n' prime numbers

    This program may be done using functions
3. WAP to create a pyramid of the character '*' and a reverse pyramid

```
    *
   ***
  *****
 *******
*********
```

```
*********
 *******
  *****
   ***
    *
```

4. WAP that accepts a character and performs the following:
    a. print whether the character is a letter or numeric digit or a special character
    b. if the character is a letter, print whether the letter is uppercase or lowercase
    c. if the character is a numeric digit, prints its name in text (e.g., if input is 9, output is NINE)
5. WAP to perform the following operations on a string
    a. Find the frequency of a character in a string.
    b. Replace a character by another character in a string.
    c. Remove the first occurrence of a character from a string.
    d. Remove all occurrences of a character from a string. WAP to swap the first n characters of two strings.
6. Write a function that accepts two strings and returns the indices of all the occurrences of the second string in the first string as a list. If the second string is not present in the first string then it should return -1.
7. WAP to create a list of the cubes of only the even integers appearing in the input list (may have elements of other types also) using the following:
    a. 'for' loop
    b. list comprehension

8. WAP to read a file and

   a. Print the total number of characters, words and lines in the file.
   b. Calculate the frequency of each character in the file. Use a variable of dictionary type to maintain the count.
   c. Print the words in reverse order.
   d. Copy even lines of the file to a file named 'File1' and odd lines to another file named 'File2'.

9. Define a class *Employee* that stores information about employees in the company. The class should contain the following:

   (1) data members- count (to keep a record of all the objects being created for this class) and for every employee: an employee number, Name, Dept, Basic, DA and HRA.

   (2) function members:

      a. __init__ method to initialize and/or update the members. Add statements to ensure that the program is terminated if any of Basic, DA and HRA is set to a negative value.
      b. function salary, that returns salary as the sum of Basic, DA and HRA.
      c. __del__ function to decrease the number of objects created for the class
      d. __str __ function to display the details of an employee along with the salary of an employee in a proper format.

10. Write a program to define a class "2DPoint" with coordinates x and y as attributes. Create relevant methods and print the objects. Also define a method distance to calculate the distance between any two point objects.

11. Write a function that prints a dictionary where the keys are numbers between 1 and 5 and the values are cubes of the keys.

12. Inherit the above class to create a "3Dpoint" with additional attribute z. Override the method defined in "2DPoint" class , to calculate distance between two points of the " 3DPoint" class.

13. Consider a tuple t1=(1, 2, 5, 7, 9, 2, 4, 6, 8, 10). WAP to perform following operations:

   a. Print half the values of the tuple in one line and the other half in the next line.

b. Print another tuple whose values are even numbers in the given tuple.

c. Concatenate a tuple t2=(11,13,15) with t1.

d. Return maximum and minimum value from this tuple

14. WAP to accept a name from a user. Raise and handle appropriate exception(s) if the text entered by the user contains digits and/or special characters.

DISCIPLINE SPECIFIC ELECTIVE COURSE: Data Mining -

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	**4**	**3**	**0**	**1**	Pass in Class XII	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:

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

SYLLABUS OF DSE

Unit 1 (7 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 (8 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, Distance-based method: K-means algorithm, measures for cluster validation, determine optimal number of clusters. Density-Based Method: DBSCAN Algorithm, Comparison of these two methods

Unit 4 (8 hours)

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

Unit 5 (11 hours)

Classification: Naive bayes classifier, nearest neighbour classifier, decision tree, overfitting, confusion matrix, evaluation metrics and model evaluation

Text Book:

1. Tan P.N., Steinbach M, Karpatne A. and Kumar V. Introduction to Data Mining, Second edition, Sixth Impression, Pearson, 2023.

Additional References:

1. Han J., Kamber M. and Pei J. *Data Mining: Concepts and Techniques*, 3rd edition, 2011, Morgan Kaufmann Publishers.
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

Datasets may be downloaded from :

1. https://archive.ics.uci.edu/datasets
2. https://www.kaggle.com/datasets?fileType=csv
3. https://data.gov.in/
4. https://ieee-dataport.org/datasets

Suggested Practical Exercises

1. Apply data cleaning techniques on any dataset (e.g., Paper Reviews dataset in UCI repository). Techniques may include handling missing values, outliers and 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 item sets 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 to build classifiers on any two datasets. Pre-process the datasets using techniques specified in Q2. Compare the Accuracy, Precision, Recall and F1 measure reported for each dataset using the abovementioned classifiers under the following situations:
   i. Using Holdout method (Random sampling):
      a) Training set = 80% Test set = 20%
      b) Training set = 66.6% (2/3rd of total), Test set = 33.3%

ii. Using Cross-Validation:
    a) 10-fold
    b) 5-fold

5. Apply simple K-means algorithm for clustering any dataset. Compare the performance of clusters by varying the algorithm parameters. For a given set of parameters, plot a line graph depicting MSE obtained after each iteration.
6. Perform density-based clustering algorithm on a downloaded dataset and evaluate the cluster quality by changing the algorithm's parameters

Project: *Students should be promoted to take up one project on using dataset downloaded from any of the websites given above and the 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.*