UNIVERSITY OF DELHI

CNC-II/093/1(23)/2022-23/457
Dated: 14.03.2023

N O T I F I C A T I O N

Sub: Amendment to Ordinance V

[E.C Resolution No. 38-1/ (38-1-12) dated 08.12.2022]

Following addition be made to Appendix-II-A to the Ordinance V (2-A) of the Ordinances of the University;

Add the following:

Syllabi of Semester-II of the following vocational courses based on Undergraduate Curriculum Framework -2022 under Ramanujan College, Jesus & Mary College & Kalindi College to be implemented from the Academic Year 2022-23.

B.Voc.– Software Development (Ramanujan College)

Category-I

B.VOC - Software Development course for Undergraduate Programme of study with Software Development as a Single Core Discipline

DISCIPLINE SPECIFIC CORE COURSE – 4: Database Management 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		
Database Management Systems	4	2	0	2	Class XII pass with Mathematics	NIL

Learning Objectives:

1. *To introduce the fundamentals of database management system and its architecture.*
2. *Students will learn about the importance of database structure and it's designing using conceptual approach using Entity Relationship Model and formal approach using Normalization.*
3. *The course would give students hands-on practice of structured query language in a relational database management system.*

Learning Outcomes:

1. Use database management system software to create and manipulate the database.
2. Create conceptual data models using entity relationship diagrams for modeling real-life situations and designing the database schema.
3. Use the concept of functional dependencies to remove redundancy and update anomalies.
4. Apply normalization theory to get a normalized database scheme.
5. Write queries using relational algebra, a procedural language.
6. Implement relational databases and formulate queries to get solutions of a broad range of data retrieval and data update problems using SQL.

Unit I **(6 Hours)**

Introduction to Database: Purpose of database system, Characteristics of database approach, data models, database management system, database system architecture, three-schema architecture, components of DBMS, data independence, and file system approach vs. database system approach.

Unit II **(6 Hours)**

Entity Relationship Modeling: Conceptual data modeling - motivation, entities, entity types, attributes, relationships, relationship types, constraints on relationship, Entity Relationship diagram notation.

Unit III **(6 Hours)**

Relational Data Model: Update anomalies, Relational Data Model - Concept of relations, schema-instance distinction, keys, relational integrity constraints, referential integrity and foreign keys, relational algebra operators and queries.

Unit IV **(6 Hours)**

Structured Query Language (SQL): Querying in SQL, DDL to create database and tables, table constraints, update database-update behaviors, DML, aggregation functions group by and having clauses, retrieve data from the database, generate and query views. Access and manipulate databases using ODBC. Basic Database administration SQL commands.

Unit V **(6 Hours)**

Database Design: Mapping an Entity Relationship model to relational database, functional dependencies and Normal forms, 1NF, 2NF, 3NF and BCNF decompositions and desirable properties of them.

Essential/Recommended readings:

1. *Elmasri, R., Navathe, B. S. Fundamentals of Database Systems, 7th Edition, Pearson Education, 2015.*
2. *Krogh, J. W. MySQL Connector/Python Revealed: SQL and NoSQL Data Storage Using MySQL for*

Python Programmers, Apress, 2018.
3. Murach J. Murach's MySQL, 3th Edition, Pearson, 2019.

Practical Component: (60 Hours)

I. Create and use the following student-society database schema for a college to answer the given (sample) queries using the standalone SQL editor.

STUDENT	Roll No	StudentName	Course	DOB
	Char(6)	Varchar(20)	Varchar(10)	Date

SOCIETY	SocID	SocName	MentorName	TotalSeats
	Char(6)	Varchar(20)	Varchar(15)	Unsigned int

ENROLLMENT	Roll No	SID	DateOfEnrollment
	Char(6)	Char(6)	Date

Here Rollno (ENROLLMENT) and SID (ENROLLMENT) are foreign keys.

1. Retrieve names of students enrolled in any society.
2. Retrieve all society names.
3. Retrieve students' names starting with letter 'A'.
4. Retrieve students' details studying in courses 'computer science' or 'chemistry'.
5. Retrieve students' names whose roll no either starts with 'X' or 'Z' and ends with '9'
6. Find society details with more than N TotalSeats where N is to be input by the user
7. Update society table for mentor name of a specific society
8. Find society names in which more than five students have enrolled
9. Find the name of youngest student enrolled in society 'NSS'
10. Find the name of most popular society (on the basis of enrolled students)
11. Find the name of two least popular societies (on the basis of enrolled students)
12. Find the student names who are not enrolled in any society
13. Find the student names enrolled in at least two societies
14. Find society names in which maximum students are enrolled
15. Find names of all students who have enrolled in any society and society names in which at least one student has enrolled
16. Find names of students who are enrolled in any of the three societies 'Debating', 'Dance' and 'Sashakt'.
17. Find society names such that its mentor has a name with 'Gupta' in it.
18. Find the society names in which the number of enrolled students is only 10% of its capacity.
19. Display the vacant seats for each society.
20. Increment Total Seats of each society by 10%
21. Add enrollment fees paid ('yes'/'No') field in the enrollment table.
22. Update date of enrollment of society id 's1' to '2018-01-15', 's2' to current date and 's3' to '2018-01-02'.
23. Create a view to keep track of society names with the total number of students enrolled in it.
24. Find student names enrolled in all the societies.

25. Count number of societies with more than 5 student enrolled in it
26. Add column Mobile number in student table with default value '9999999999'
27. Find the total number of students whose age is > 20 years.
28. Find names of students who are born in 2001 and are enrolled in at least one society.
29. Count all societies whose name starts with 'S' and ends with't' and at least 5 students are enrolled in the society.
30. Display the following information:

Society name, Mentor name, Total Capacity, Total Enrolled, Unfilled Seats

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

DISCIPLINE SPECIFIC CORE COURSE – 5: Programming in JAVA

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		
Programming in JAVA	**4**	**2**	**0**	**2**	**Class XII pass with Mathematics**	**DSC -1**

Learning Objectives:

1. *To develop structured as well as object-oriented programming skills using JAVA Programming.*
2. *The course provides a complete understanding of the object-oriented programming features, namely Encapsulation, Abstraction, Inheritance and Polymorphism.*

Learning Outcomes:

1. *Implement Exception Handling and File Handling.*
2. *Implement multiple inheritance using Interfaces.*
3. *Logically organize classes and interfaces using packages.*
4. *Use AWT and Swing to design GUI applications.*

Unit I **(4 Hours)**

Review of Object Oriented Programming and Java Fundamentals: Structure of Java programs, Classes and Objects, Data types, Type Casting, Looping Constructs.

Unit II **(6 Hours)**

Interfaces Interface basics; Defining, implementing and extending interfaces; Implementing multiple inheritance using interfaces Packages Basics of packages, Creating and accessing packages, System packages, Creating user defined packages

Unit III **(4 Hours)**

Exception handling using the main keywords of exception handling: try, catch, throw, throws and finally; Nested try, multiple catch statements, creating user defined exceptions

Unit IV **(4 Hours)**

File Handling Byte Stream, Character Stream, File I/O Basics, File Operations.

Unit V **(6 Hours)**

AWT and Event Handling: The AWT class hierarchy, Events, Event sources, Event classes, Event Listeners, Relationship between Event sources and Listeners, Delegation event model, Creating GUI applications using AWT.

Unit VI **(6 Hours)**

Swing Introduction to Swing, Swing vs. AWT, and Hierarchy for Swing components, Creating GUI, Applications using Swing.

Essential/Recommended readings:

1. *Schildt, H. (2018). Java: The Complete Reference. 10th edition. McGraw-Hill Education.*
2. *Horstmann, C. S. (2017). Core Java - Vol. I – Fundamentals (Vol. 10). Pearson Education.*

List of Practical(60 Hours)

1. Design a class Complex having a real part (x) and an imaginary part (y). Provide methods to perform the following on complex numbers:
    a. Add and Multiply two complex numbers.
    b. toString() method to display complex numbers in the form: x + i y
2. Create a class TwoDim which contains private members as x and y coordinates in package P1. Define the default constructor, a parameterized constructor and override toString() method to display the co-ordinates. Now reuse this class and in package P2 create another class ThreeDim, adding a new dimension as z as its private member.
Define the constructors for the subclass and override toString() method in the subclass also. Write appropriate methods to show dynamic method dispatch. The main() function should be in a package P.
3. Define an abstract class Shape in package P1. Inherit two more classes: Rectangle in package P2 and Circle in package P3. Write a program to ask the user for the type of shape and then using the concept of dynamic method dispatch, display the area of the appropriate subclass. Also write appropriate methods to read the data. The main() function should not be in any package.
4. Create an exception subclass UnderAge, which prints “Under Age” along with the age value when an object of UnderAge class is printed in the catch statement. Write a class exceptionDemo in which the method test() throws UnderAge exception if the variable age passed to it as argument is less than 18. Write main() method also to show working of the program.
5. Write a program to implement stack. Use exception handling to manage underflow and overflow conditions.
6. Write a program that copies content of one file to another. Pass the names of the files through command-line arguments.
7. Write a program to read a file and display only those lines that have the first two characters as '//' (Use try with resources).
8. Write a program to create an Applet. Create a frame as a child of applet. Implement mouseClicked(), mouseEntered( ) and mouseExited( ) events for applet. Frame is visible when mouse enters applet window and hidden when mouse exits from the applet window.
9. Write a program to display a string in frame window with pink color as background.
10. Write a program to create an Applet that has two buttons named “Red” and “Blue”. When a button is pressed the background color of the applet is set to the color named by the button’s label.
11. Create an applet which responds to KEY_TYPED event and updates the status window with message (“Typed character is: X”). Use adapter class for other two events.
12. Create an applet with two buttons labeled ‘A’ and ‘B’. When button ‘A’ is pressed, it displays your personal information (Name, Course, Roll No, and College) and when button ‘B’ is pressed, it displays your CGPA in previous semester.
13. Write a program that creates a Banner and then creates a thread to scrolls the message in the banner from left to right across the applet’s window.
14. Rewrite the applet programs using Swing.

DISCIPLINE SPECIFIC CORE COURSE – 6: Mathematics for Computing - 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		
Mathematics for Computing - II	**4**	**3**	**1**	**0**	**Class XII pass with Mathematics**	**DSC - 3**

Learning Objectives:

1. *To study the fundamental concepts and topics of probability and statistics.*
2. *The study of this course is important for students to learn machine learning and similar courses in later semesters.*

Learning Outcomes:

1. Use probability theory to evaluate the probability of real-world events.
2. Describe discrete and continuous probability distribution functions and generate random numbers from the given distributions.
3. Find the distance between two probability distributions
4. Define and quantify the information contained in the data.
5. Perform data analysis in a probabilistic framework.
6. Visualize and model the given problem using mathematical concepts covered in the course.

Unit I **(9 Hours)**

Basic Probability: Introduction to the notion of probability, Random experiment, Sample space and Events, Probability defined on events, Algebra of events. Conditional probabilities, independent events, Bayes' theorem.

Unit II **(12 Hours)**

Random Variables: Introduction to Random Variables, Probability mass/density functions, Cumulative distribution functions. Discrete Random Variables (Bernoulli, Binomial, Poisson, Multinomial and Geometric). Continuous Random Variables (Uniform, Exponential and Normal). Expectation of a Random Variable, Expectation of Function of a Random Variable and Variance. Markov inequality, Chebyshev's inequality, Central Limit Theorem, Weak and Strong Laws of Large Numbers.

Unit III **(12 Hours)**

Joint Distributions: Jointly distributed Random Variables, Joint distribution functions, Independent Random Variables, Covariance of Random Variables, Correlation Coefficients, Conditional Expectation.

Unit IV **(12 Hours)**

Markov Chain and Information Theory: Introduction to Stochastic Processes, Chapman–Kolmogorov equations, Classification of states, Limiting and Stationary Probabilities. Random Number Generation, Pseudo Random Numbers, Inverse Transformation Method, Rejection Method, Uncertainty, Information and Entropy, Mutual Information, KL Divergence.

Essential/Recommended readings:

1. *Sheldon Ross, Introduction to Probability Models, 12th Edition, Elsevier, 2019.*
2. *K.S. Trivedi, Probability and Statistics with Reliability, Queuing and Computer Science Applications, 2nd Edition, Wiley, 2015.*
3. *Marc Peter Deisenroth, A. Aldo Faisal and Cheng Soon Ong, Mathematics for Machine Learning, 1st Edition, Cambridge University Press, 2020.*
4. *Ian F. Blake, "An Introduction to Applied Probability", John Wiley.*

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