



ST. FRANCIS DE SALES COLLEGE

A FRANSALIAN INSTITUTE OF HIGHER EDUCATION **AUTONOMOUS**

NAAC A GRADE • AFFILIATED TO BANGALORE UNIVERSITY • AICTE APPROVED • 2(F) & 12 (B) RECOGNITION OF UGC • ISO 9001:2015 CERTIFIED

📍 Electronics City P.O., Bengaluru - 560 100, Karnataka, INDIA 📞 (+91) 8088140679 📧 pro@sfscollge.in 🌐 www.sfscollge.in

BCA [3 YEARS] Syllabus

and

Scheme From 2024-2025

**BOARD OF STUDIES
[COMPUTER SCIENCE AND APPLICATIONS]**

**St. Francis de Sales College [Autonomous]
Electronics City P.O. Bengaluru 560100 Karnataka,
INDIA**

TABLE OF CONTENTS

Sl. No.	PARTICULARS	PAGE NUMBER
1.	Members of the Board of Studies	1
2.	Preface to the BCA 2024-2025 SEP Syllabus	3
3.	St. Francis de Sales College (Autonomous) - Vision and Mission	5
4.	Department of Computer Science and Applications - Vision and Mission	6
5.	Eligibility Criteria	7
6.	Programme Structure and Duration	7
7.	Promotion	7
8.	Programme Outcome (PO)	8
9.	Continuous Internal Assessment Criteria	9
10.	Grading System and Grade Description	10
11.	External Evaluation - Theory course	11
12.	Course Matrix as per SEP 2024	12
13.	24BCA13 Discrete Structures	13
14.	24BCA14 Problem Solving Techniques	15
15.	24BCA15 Computer Architecture	17
16.	24BCA16 Problem Solving Techniques Lab	19
17.	24BCA17Computer Architecture Lab	21
18.	24BCA18 Office Automation Tools Lab	23
19.	24BCA23 Data Structure	25
20.	24BCA24 Object-Oriented Programming Using JAVA	27
21.	24BCA25 Operating Systems	29
22.	24BCA26 Data Structure Lab	31
23.	24BCA27 Object Oriented Programming Lab	33
24.	24BCA28 LINUX and Shell Programming Lab	35

MEMBERS OF THE BOARD OF STUDIES

Sl. No.	Name	Designation
1.	Dr. S. Sivagami, Program In-charge and Assistant Professor, St. Francis de Sales College (Autonomous), Electronic City, Bengaluru.	Chairperson
2.	Dr. Hanumanthappa M Senior Professor & Director, ISC & PM- USHA project, Ex- Registrar, Department of Computer Science, Bangalore University, Bangalore.	University Nominee
3.	Dr. Sabeen Govind P V Assistant Professor, Rajagiri College of Social Sciences	External Expert
4.	Dr. Kousalya Govardhanan Professor & Dean of Research, Dayananda Sagar University, Electronic City (Subject Expert nominated by the Academic Council)	External Expert
5.	Mr. Maria Joseph Frederic Senior Manager, IBM ISL R&D	Industry Expert
6.	Mr. Phani Pramod, Senior Development Manager, Essbase and Database tools, Oracle	Industry Expert
7.	Ms. Umme Hermain Shaikh Associate Consultant, Tarento Technologies, Bengaluru	Alumni
8.	Ms. Sailaja M Assistant Professor, St. Francis de Sales College (Autonomous), Electronic City, Bengaluru.	Member
9.	Ms. S. Annie Christella Assistant Professor, St. Francis de Sales College (Autonomous), Electronic City, Bengaluru.	Member

10.	Ms. Saranya C Assistant Professor, St. Francis de Sales College (Autonomous), Electronic City, Bengaluru.	Member
-----	---	--------

11.	Ms. Thejaswi Nandyala Assistant Professor, St. Francis de Sales College (Autonomous), Electronic City, Bengaluru.	Member
12.	Ms. Amruta Gadad Assistant Professor, St. Francis de Sales College (Autonomous), Electronic City, Bengaluru.	Member
13.	Ms. Sathiya Priya Assistant Professor, St. Francis de Sales College (Autonomous), Electronic City, Bengaluru.	Member
14.	Ms. Gowthami Gunasekar Assistant Professor, St. Francis de Sales College (Autonomous), Electronic City, Bengaluru.	Member
15.	Ms. Arundhati Ghosh, Assistant Professor, St. Francis de Sales College (Autonomous), Electronic City, Bengaluru.	Member
16.	Mr. Joseph Rajakumar, Assistant Professor, St. Francis de Sales College (Autonomous), Electronic City, Bengaluru.	Member
17.	Mr. Kirubakaran, Assistant Professor, St. Francis de Sales College (Autonomous), Electronic City, Bengaluru.	Member
18.	Ms. Samadrita Chakraborty, Assistant Professor, St. Francis de Sales College (Autonomous), Electronic City, Bengaluru.	Member

PREFACE TO THE BCA 2024-2025 SYLLABUS

ಕರ್ನಾಟಕ ಸರ್ಕಾರದ ನಡಾವಳಿಗಳು

ವಿಷಯ: ಉನ್ನತ ಶಿಕ್ಷಣ ಇಲಾಖೆಯ ವ್ಯಾಪ್ತಿಯಲ್ಲಿನ ವಿಶ್ವವಿದ್ಯಾಲಯಗಳು ಮತ್ತು ಕಾಲೇಜುಗಳಲ್ಲಿ 2024-25ನೇ ಸಾಲಿನಿಂದ ಪದವಿ ಕಾರ್ಯಕ್ರಮಗಳ ಅವಧಿ ಮತ್ತು ಪರೀಕ್ಷಾಮುಖಾಂತರ ಪರಿಷ್ಕರಿಸುವ ಬಗ್ಗೆ.

- ಓದಲಾಗಿದೆ:
1. ಸರ್ಕಾರದ ಆದೇಶ ಸಂಖ್ಯೆ: ಇಡಿ 260 ಯುಎನ್‌ಇ 2019 (ಭಾಗ-1) ಬೆಂಗಳೂರು, ದಿನಾಂಕ:07.08.2021.
 2. ಸರ್ಕಾರಿ ಆದೇಶ ಸಂಖ್ಯೆ:ಇಡಿ 166 ಯುಎನ್‌ಇ 2023, ಬೆಂಗಳೂರು ದಿನಾಂಕ:11.10.2023
 3. ರಾಜ್ಯ ಶಿಕ್ಷಣ ನೀತಿ ಆಯೋಗದ ಮಧ್ಯಂತರ ವರದಿ ದಿನಾಂಕ:18.01.2024.
 4. ಕರ್ನಾಟಕ ರಾಜ್ಯ ಉನ್ನತ ಶಿಕ್ಷಣ ಪರಿಷತ್ತಿನ ಟಿಪ್ಪಣಿ ದಿನಾಂಕ: 14.02.2024.
 5. ರಾಜ್ಯ ಶಿಕ್ಷಣ ನೀತಿ ಆಯೋಗದ ವರದಿಗಳ ದಿನಾಂಕ: 19.01.2024 ಮತ್ತು 10.03.2024.
 6. ಸರ್ಕಾರದ ಮುಖ್ಯ ಕಾರ್ಯದರ್ಶಿಯವರ ಕಛೇರಿಯ ಅನಧಿಕೃತ ಟಿಪ್ಪಣಿ ಸಂಖ್ಯೆ: CS/05/SCM/2024, dated: 02.04.2024.

ಮೇಲೆ ಓದಲಾದ ಕ್ರಮ ಸಂಖ್ಯೆ (1) ರ ದಿನಾಂಕ: 07.08.2021ರ ಆದೇಶದಲ್ಲಿ ಭಾರತ ಸರ್ಕಾರವು ಪ್ರಕಟಿಸಿರುವ ರಾಷ್ಟ್ರೀಯ ಶಿಕ್ಷಣ ನೀತಿಯನ್ನು ರಾಜ್ಯದ ಉನ್ನತ ಶಿಕ್ಷಣ ಇಲಾಖೆಯಡಿ ವಿಶ್ವವಿದ್ಯಾಲಯಗಳು ಮತ್ತು ಸಂಯೋಜಿತ ಕಾಲೇಜುಗಳಲ್ಲಿ 2021-22ನೇ ಶೈಕ್ಷಣಿಕ ವರ್ಷದಿಂದ ಅಗತ್ಯ ಮಾರ್ಗ ಸೂಚಿಗಳನುಸಾರ ಅನುಷ್ಠಾನಗೊಳಿಸಲಾಗಿರುತ್ತದೆ.

ಮೇಲೆ ಓದಲಾದ ಕ್ರಮ ಸಂಖ್ಯೆ (2) ರ ದಿನಾಂಕ: 11.10.2023ರ ಆದೇಶದಲ್ಲಿ ಹೊಸ ಶಿಕ್ಷಣ ನೀತಿಯನ್ನು ರೂಪಿಸುವ ಉದ್ದೇಶದಿಂದ ಕರ್ನಾಟಕ ರಾಜ್ಯ ಶಿಕ್ಷಣ ನೀತಿಯ ಕರಡನ್ನು ಸಿದ್ಧಪಡಿಸಲು ಶ್ರೀಷ್ಠ ಶಿಕ್ಷಣ ತಜ್ಞರಾದ Prof. Sukhdev Torat, ಇವರ ಅಧ್ಯಕ್ಷತೆಯಲ್ಲಿ ರಾಜ್ಯ ಶಿಕ್ಷಣ ನೀತಿ ಆಯೋಗವನ್ನು ರಚಿಸಲಾಗಿರುತ್ತದೆ.

ಮೇಲೆ ಓದಲಾದ ಕ್ರಮ ಸಂಖ್ಯೆ (3) ರಲ್ಲಿ ರಾಜ್ಯ ಶಿಕ್ಷಣ ನೀತಿ ಆಯೋಗವು ದಿನಾಂಕ: 18.01.2024ರಂದು ಸರ್ಕಾರಕ್ಕೆ ಮಧ್ಯಂತರ ವರದಿಯನ್ನು ಸಲ್ಲಿಸಿರುತ್ತದೆ.

ಮೇಲೆ ಓದಲಾದ ಕ್ರಮ ಸಂಖ್ಯೆ (4) ರಲ್ಲಿ ಕರ್ನಾಟಕ ರಾಜ್ಯ ಉನ್ನತ ಶಿಕ್ಷಣ ಪರಿಷತ್ತು ರಾಜ್ಯ ಶಿಕ್ಷಣ ನೀತಿ ಆಯೋಗವು ಸರ್ಕಾರಕ್ಕೆ ಸಲ್ಲಿಸಿರುವ ಮಧ್ಯಂತರ ವರದಿಯನ್ನು ಸಂಪೂರ್ಣವಾಗಿ ಒಪ್ಪಲು ಶಿಫಾರಸ್ಸು ಮಾಡಿರುತ್ತದೆ.

ಮೇಲೆ ಓದಲಾದ ಕ್ರಮ ಸಂಖ್ಯೆ (5) ರಲ್ಲಿ ರಾಜ್ಯ ಶಿಕ್ಷಣ ನೀತಿ ಆಯೋಗವು ದಿನಾಂಕ: 19.01.2024ರಂದು REPORT PART-1 ಮತ್ತು ದಿನಾಂಕ: 10.03.2024ರಂದು ನ್ನು REPORT PART-1(a) ಮಧ್ಯಂತರ ವರದಿಗಳನ್ನು ಸರ್ಕಾರಕ್ಕೆ ಸಲ್ಲಿಸಿರುತ್ತದೆ.

ಮೇಲೆ ಓದಲಾದ ಕ್ರಮ ಸಂಖ್ಯೆ (6) ರಲ್ಲಿ ರಾಜ್ಯ ಶಿಕ್ಷಣ ನೀತಿ ಆಯೋಗವು ಸರ್ಕಾರಕ್ಕೆ ಸಲ್ಲಿಸಿರುವ ಮಧ್ಯಂತರ ವರದಿಯಲ್ಲಿನ ಶಿಫಾರಸ್ಸುಗಳನ್ನು 2024-25 ನೇ ಶೈಕ್ಷಣಿಕ ಸಾಲಿನಿಂದ ಪದವಿ ಕಾರ್ಯಕ್ರಮಗಳಿಗೆ ಅನುಷ್ಠಾನಗೊಳಿಸಲು ದಿನಾಂಕ: 02.04.2024ರಂದು ಚುನಾವಣಾ ಆಯೋಗದ ಅನುಮತಿಯನ್ನು ಪಡೆಯಲಾಗಿರುತ್ತದೆ.

ಮೇಲ್ಕಂಡ ಅಂಶಗಳ ಹಿನ್ನೆಲೆಯಲ್ಲಿ, ರಾಜ್ಯ ಶಿಕ್ಷಣ ನೀತಿ ಆಯೋಗವು ಸರ್ಕಾರಕ್ಕೆ ಸಲ್ಲಿಸಿರುವ ಮಧ್ಯಂತರ ವರದಿಗಳಲ್ಲಿನ ಶಿಫಾರಸ್ಸುಗಳನ್ನು 2024-25 ನೇ ಶೈಕ್ಷಣಿಕ ಸಾಲಿನಿಂದ ಪದವಿ ಕಾರ್ಯಕ್ರಮಗಳಿಗೆ ಅನುಷ್ಠಾನಗೊಳಿಸಲು ಉದ್ದೇಶಿಸಿ, ಸರ್ಕಾರದ ಆದೇಶ ಸಂಖ್ಯೆ: ಇಡಿ 260 ಯುಎನ್‌ಇ 2019 (ಭಾಗ-1)

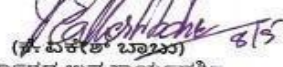
ಬೆಂಗಳೂರು, ದಿನಾಂಕ:07.08.2021ರ ಆದೇಶವನ್ನು ಪರಿಷ್ಕರಿಸಲು ಸರ್ಕಾರವು ನಿರ್ಧರಿಸಿ, ಅದರಂತೆ, ಈ ಕೆಳಕಂಡ ಆದೇಶ.

ಸರ್ಕಾರದ ಆದೇಶ ಸಂಖ್ಯೆ: ಇಡಿ 166 ಯುಎನ್ಇ 2023, ಬೆಂಗಳೂರು, ದಿನಾಂಕ: 08.05.2024.

ಪ್ರಸ್ತಾವನೆಯಲ್ಲಿ ವಿವರಿಸಿರುವ ಅಂಶಗಳನ್ವಯ ರಾಜ್ಯದಲ್ಲಿ ಉನ್ನತ ಶಿಕ್ಷಣ ಇಲಾಖೆಯಡಿಯ ವಿಶ್ವವಿದ್ಯಾಲಯಗಳು ಮತ್ತು ಸಂಯೋಜಿತ ಕಾಲೇಜುಗಳಲ್ಲಿ ಪದವಿ ಕಾರ್ಯಕ್ರಮಗಳಿಗೆ 2024-25ನೇ ಶೈಕ್ಷಣಿಕ ವರ್ಷದಿಂದ ಅನುಬಂಧ-1 ಮತ್ತು ಅನುಬಂಧ-2 ರ ಮಾರ್ಗಸೂಚಿಗಳಿಗನುಸಾರವಾಗಿ ಪದವಿ ಕಾರ್ಯಕ್ರಮಗಳ ಅವಧಿ ಮತ್ತು ಪಠ್ಯಕ್ರಮವನ್ನು ಪರಿಷ್ಕರಿಸಿ ಅನುಷ್ಠಾನಗೊಳಿಸಿ ಆದೇಶಿಸಲಾಗಿದೆ.

1. ಅನುಷ್ಠಾನದ ಮಾರ್ಗಸೂಚಿಗಳು - ಅನುಬಂಧ-1
2. ಪ್ರೋಗ್ರಾಮ್ ವಿನ್ಯಾಸ (Curriculum Structure) - ಅನುಬಂಧ-2

ಕರ್ನಾಟಕ ರಾಜ್ಯಪಾಲರ ಆಜ್ಞಾನುಸಾರ
ಮತ್ತು ಅವರ ಹೆಸರಿನಲ್ಲಿ



ಸರ್ಕಾರದ ಉಪ ಕಾರ್ಯದರ್ಶಿ

ಉನ್ನತ ಶಿಕ್ಷಣ ಇಲಾಖೆ (ವಿಶ್ವವಿದ್ಯಾಲಯಗಳು)

ಇವರಿಗೆ,

1. ಸರ್ಕಾರದ ಮುಖ್ಯ ಕಾರ್ಯದರ್ಶಿರವರ ಆಪ್ತ ಕಾರ್ಯದರ್ಶಿ, ಕರ್ನಾಟಕ ಸರ್ಕಾರ.
2. ಕುಲಪತಿಗಳು, ಉನ್ನತ ಶಿಕ್ಷಣ ಇಲಾಖೆಯಡಿಯ ಸಾರ್ವಜನಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯಗಳು, ಕರ್ನಾಟಕ.
3. ಸರ್ಕಾರದ ಅಪರ ಮುಖ್ಯ ಕಾರ್ಯದರ್ಶಿಗಳು, ಮಾನ್ಯ ಮುಖ್ಯ ಮಂತ್ರಿಗಳ ಕಛೇರಿ, ವಿಧಾನಸೌಧ, ಬೆಂಗಳೂರು.
4. ಸರ್ಕಾರದ ಅಪರ ಮುಖ್ಯ ಕಾರ್ಯದರ್ಶಿಗಳು ಮತ್ತು ಅಭಿವೃದ್ಧಿ ಆಯುಕ್ತರು, ವಿಧಾನಸೌಧ, ಬೆಂಗಳೂರು.
5. ಸರ್ಕಾರದ ಅಪರ ಮುಖ್ಯ ಕಾರ್ಯದರ್ಶಿಗಳು, ಆರ್ಥಿಕ ಇಲಾಖೆ, ವಿಧಾನಸೌಧ, ಬೆಂಗಳೂರು.
6. ಸರ್ಕಾರದ ಅಪರ ಮುಖ್ಯ ಕಾರ್ಯದರ್ಶಿಗಳು, ಸಿಸುಆಇ (ಇ-ಆಡಳಿತ ಇಲಾಖೆ), ವಿಧಾನಸೌಧ, ಬೆಂಗಳೂರು.
7. ಸರ್ಕಾರದ ಪ್ರಧಾನ ಕಾರ್ಯದರ್ಶಿ, ಶಾಲಾ ಶಿಕ್ಷಣ ಮತ್ತು ಸಾಕ್ಷರತಾ ಇಲಾಖೆ, ಬಹುಮಹಡಿ ಕಟ್ಟಡ, ಬೆಂಗಳೂರು.
8. ಆಯುಕ್ತರು, ಕಾಲೇಜು ಮತ್ತು ತಾಂತ್ರಿಕ ಶಿಕ್ಷಣ ಇಲಾಖೆ, ಜೇಷಾಧಿ ರಸ್ತೆ, ಬೆಂಗಳೂರು-1.
9. ಉಪಾಧ್ಯಕ್ಷರು, ಕರ್ನಾಟಕ ರಾಜ್ಯ ಉನ್ನತ ಶಿಕ್ಷಣ ಪರಿಷತ್, ಬೆಂಗಳೂರು.
10. ಕಾರ್ಯ ನಿರ್ವಾಹಕ ನಿರ್ದೇಶಕರು, ಕರ್ನಾಟಕ ರಾಜ್ಯ ಉನ್ನತ ಶಿಕ್ಷಣ ಪರಿಷತ್, ಬೆಂಗಳೂರು.
11. ಕುಲಸಚಿವರು (ಆಡಳಿತ) ಉನ್ನತ ಶಿಕ್ಷಣ ಇಲಾಖೆಯಡಿಯಲ್ಲಿನ ಸಾರ್ವಜನಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯಗಳು, ಕರ್ನಾಟಕ.
12. ಕುಲಸಚಿವರು (ಮೌಲ್ಯಮಾಪನ), ಉನ್ನತ ಶಿಕ್ಷಣ ಇಲಾಖೆಯಡಿಯಲ್ಲಿನ ಸಾರ್ವಜನಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯಗಳು, ಕರ್ನಾಟಕ.
13. ಮಾನ್ಯ ಉನ್ನತ ಶಿಕ್ಷಣ ಸಚಿವರ ಆಪ್ತ ಕಾರ್ಯದರ್ಶಿ, ಉನ್ನತ ಶಿಕ್ಷಣ ಇಲಾಖೆ, ವಿಧಾನಸೌಧ, ಬೆಂಗಳೂರು.
14. ಮಾನ್ಯ ಶಾಲಾ ಶಿಕ್ಷಣ ಮತ್ತು ಸಾಕ್ಷರತಾ ಸಚಿವರ ಆಪ್ತ ಕಾರ್ಯದರ್ಶಿ, ವಿಧಾನಸೌಧ, ಬೆಂಗಳೂರು.
15. ಸರ್ಕಾರದ ಅಪರ ಮುಖ್ಯ ಕಾರ್ಯದರ್ಶಿರವರ ಆಪ್ತ ಕಾರ್ಯದರ್ಶಿ, ಉನ್ನತ ಶಿಕ್ಷಣ ಇಲಾಖೆ, ವಿಧಾನಸೌಧ, ಬೆಂಗಳೂರು.
16. ಸರ್ಕಾರದ ಉಪ ಕಾರ್ಯದರ್ಶಿ (ವಿಶ್ವವಿದ್ಯಾಲಯಗಳು), ಉನ್ನತ ಶಿಕ್ಷಣ ಇಲಾಖೆ, ಬೆಂಗಳೂರು.
17. ಸರ್ಕಾರದ ಉಪ ಕಾರ್ಯದರ್ಶಿ (ಕಾಲೇಜು ಮತ್ತು ತಾಂತ್ರಿಕ ಶಿಕ್ಷಣ), ಉನ್ನತ ಶಿಕ್ಷಣ ಇಲಾಖೆ, ಬೆಂಗಳೂರು.

Based on the order received from the parent university (Bangalore University - given above), the Board of Studies (Computer Applications) of St. Francis de Sales College (Autonomous), has decided to adopt the grading scheme of the parent university for its BCA Syllabus from 2024-2025 onwards.

ST. FRANCIS DE SALES COLLEGE (AUTONOMOUS) ABOUT THE COLLEGE

St. Francis de Sales College (Autonomous), popularly known as SFS College, is one of the leading Institutions of Higher Education in Bengaluru, Karnataka. Founded in 2004 with the vision of Excellence, Efficiency, and Transformation, and the Mission of Love of God and Service to Humanity, the College is run by the Missionaries of St. Francis de Sales (MSFS) of the South West India Province, also known as Fransalians. The College is accredited with “A” grade by NAAC, approved by AICTE, recognized under 2(f) & 12(b) by UGC, and certified under ISO 9001:2015. Permanently affiliated to Bangalore University, the College offers several degree programs at the Bachelors, Masters, and Doctoral levels under various disciplines. In 2024, St. Francis de Sales College received the Autonomous status, and it remains as a center for quality education, equipping the students with the skills, knowledge, and values needed to excel and make a meaningful impact in the world.

VISION AND MISSION VISION

Excellence, Efficiency and Transformation.

MISSION

Love of God and Service to Humanity.

DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS

The Computer Science and Applications Department is dedicated to advancing the understanding of computational systems and technologies through rigorous education, innovative research, and community engagement. The department offers a comprehensive curriculum that blends theoretical foundations with practical skills to prepare students for the rapidly evolving technology landscape. With a focus on problem-solving, software development, and cutting-edge research, the department strives to equip students with the tools and knowledge required to excel in a variety of computing fields.

VISION AND MISSION VISION

Empowering through technology, innovation and expertise

MISSION

Leveraging computation knowledge to drive societal progress and student success.

ELIGIBILITY CRITERIA

Candidates who have completed two years Pre-University course of Karnataka State or its equivalent as notified by the Government from time to time are eligible to seek admission for this programme. The students of other states and foreign countries are eligible in accordance with state and central government guidelines from time to time.

PROGRAMME STRUCTURE AND DURATION

The programme is for Three (03) years consisting of Six Semesters altogether. A candidate shall complete his/her degree within six (06) academic years from the date of his/her admission to the first semester. A Student who successfully completes Three (03) years of the programme will be awarded Bachelor's Degree in Computer Applications (B.C.A) by Bangalore University.

PROMOTION

A candidate who has obtained a minimum of 35% marks in End Semester examination and an aggregate of 40% marks in each subject shall be eligible for a pass or exemption in that subject.

PROGRAM OUTCOME (PO)

PO1	Computational Knowledge: Acquire in-depth computational and mathematical knowledge with an ability to abstract and conceptualize from defined problems and requirements.
PO2	Dynamic Problem- Solving Skill: Identify, formulate, and exhibit strong analytical and dynamic problem-solving skills to address evolving computational challenges.
PO3	Innovative System Analysis and Design/ Development: Design and evaluate solutions for complex problems in Data Science, AI & ML, and Full Stack Development, considering societal, cultural, and environmental factors.
PO4	Investigate complex computing problems: Conduct literature surveys, analyze information, and design experiments using appropriate research methods to derive valid conclusions in relevant domains.
PO5	Use of modern tools/ Adaptive proficiency: Select, adapt, and apply modern IT tools and programming languages effectively in Data Science, AI & ML, and Full Stack Development to solve diverse computing challenges.
PO6	Knowledge Optimization: Modify algorithms or software systems to improve efficiency or resource utilization.
PO7	Life Long Continuous learning and Technology Adaptability: Pursue lifelong learning to stay updated with emerging technologies in Data Science, AI & ML, and Full Stack Development for sustained employability.
PO8	Soft skills and Collaborative Teamwork: Communicate effectively, enhance interpersonal skills, and collaborate in multidisciplinary teams essential for success in professional environments.
PO9	Cyber Security Proficiency: Understand cyber threats, develop secure software, and protect sensitive data in Data Science, AI & ML, and Full Stack Development applications.
PO10	Ethical and Professional Conduct: Adhere to ethical standards and professional practices in Data Science, AI & ML, and Full Stack Development roles and responsibilities.
PO11	Employability: Identify market trends, upgrade skills accordingly, and enhance employability in Data Science, AI & ML, and Full Stack Development careers.

PO12	Innovation and Entrepreneurship: Identify opportunities, innovate, and create value through Data Science, AI & ML, and Full Stack Development projects for personal growth and societal impact.
------	---

CONTINUOUS INTERNAL ASSESSMENT

UNDERGRADUATE PROGRAMS:

THEORY:

1. Continuous Internal Assessment (C1 & C2) – 20 marks
2. End Semester Examination – 80 marks

PRACTICAL:

1. Continuous Internal Assessment (C1 & C2) – 10 marks
2. End Semester Practical Examination – 40 marks

S.NO	ASSESSMENTS	COMPONENTS	MARKS & ATTENDANCE	IA MARKS
THEORY SUBJECTS				
1	Unit Test (25% of Syllabus)	C1	25	2.5
2	Skill-Based Activities: Case Study / Seminar / Assignment / Quiz	C2	10	5
3	Mid Semester Examination (50% of Syllabus)	C2	80	5
4	Attendance ● 75.00-79.99% - 1 Mark ● 80.00-84.99% - 2 Marks ● 85.00-89.99% - 3 Marks ● 90.00-94.99% - 4 Marks ● 95.00-100.00% -5 Marks	C1	Minimum of 75%	5
Total				20 marks
SEC & PRACTICAL SUBJECTS				
1	Attendance ● 75.00-79.99% - 1 Mark ● 80.00-84.99% - 2 Marks ● 85.00-89.99% - 3 Marks ● 90.00-94.99% - 4 Marks ● 95.00-100.00% -5 Marks	C1	Minimum of 75%	5
2	Mid Semester Examination / Model Practical Examination	C2	Minimum of 75%	5
Total				10 Marks

GRADING SYSTEM

Table of Conversion of % Marks to grade point:

% Marks	Grade Point
96-100	10
91-95	9.5
86-90	9.0
81-85	8.5
76-80	8.0
71-75	7.5
66-70	7.0
61-65	6.5
56-60	6.0
51-55	5.5
46-50	5.0
41-45	4.5
40	4

Final Result/Grade Description:

Semester/ Programme % of Marks	Semester GPA/ Programme/ CGPA	Grade Alpha Sign	Result/Class Description
90.1-100	9.01-10.00	O	Outstanding
80.1-90.0	8.01-9.00	A+	First Class Exemplary
70.1-80.0	7.01-8.00	A	First Class Distinction
60.1-70.0	6.01-7.00	B+	First Class
55.1-60.0	5.51-6.00	B	High Second Class
50.1-55.0	5.01-5.50	C	Second Class
40.0-50.0	4.00-5.00	P	Pass Class
Below 40	Below 4.0	F	Re-Appear

EXTERNAL EVALUATION

THEORY COURSE

There shall be a written semester examination at the end of each semester for all theory courses of duration of 3 hours with maximum 80 marks. A question paper may contain short answer type and long essay type questions. The question paper pattern is as follows.

SECTION S	TYPE OF QUESTIONS	MARKS	NUMBER OF QUESTIONS TO BE ANSWERED
A	CONCEPTUAL	2	5 OUT OF 8
B	ANALYTICAL	5	6 OUT OF 8
C	PROBLEM SOLVING	8	5 OUT OF 6
TOTAL 80 MARKS			

DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS

COURSE MATRIX AS PER 2025

SEMESTER I

	Subjects	Paper/Subject Code	Total Teaching hrs	Duration of Exam (hrs.)	Marks			Credits
					IA	Uni. Exam	Total	
Part 2 Core Papers	Discrete Structures	24BCA13	45	4	20	80	100	4
	Problem Solving Technique	24BCA14	45	4	20	80	100	4
	Computer Architecture	24BCA15	45	4	20	80	100	4
	Problem Solving Technique Lab	24BCA16	60	3	10	40	50	2
	Computer Architecture Lab	24BCA17	60	3	10	40	50	2
	Office Automation Tools	24BCA18	60	3	10	40	50	2
Part 3 Compulsory	Environmental studies (EVS)	24EVS1X	30	1.5	20	30	50	2
	Total Credits						700	23

SEMESTER II

	Subjects	Paper/Subject Code	Total Teaching hrs	Duration of Exam (hrs.)	Marks			Credits
					IA	Uni. Exam	Total	
Part 1 Languages	Language - I:		45					
	Indian & Foreign Languages			3	20	80	100	3
	Language II: English		45	3	20	80	100	3
Part 2 Core Papers	Data Structure	24BCA23	45	4	20	80	100	4
	Object-Oriented Programming Using JAVA	24BCA24	45	4	20	80	100	4
	Operating Systems	24BCA25	45	4	20	80	100	4
Part 3 Compulsory	The Constitution of India	24BCA26	30	1.5	20	30	50	2
	Data Structure Lab	24BCA27	60	3	10	40	50	2
	Object Oriented Programming Lab	24BCA28	60	3	10	40	50	2
	LINUX and Shell Programming Lab	24BCA29	60	3	10	40	50	2
	Total Credits						700	23

SEMESTER I 24BCA13 : DISCRETE STRUCTURES

Course Code	24BCA13	Course Title	Discrete Structures	
Course Type	DSC	Contact Hours	4 Hours per Week	No of Hours: 60
Credit	4	Domain	Mathematics	
SYLLABUS				
I	Set Theory Fundamentals of Set theory, Set Operations, Laws of Set Theory, Counting and Venn Diagrams, Cartesian Product, Relations, Types of Relations, Functions, Types of Functions, Function Composition, Inverse Functions. Mathematical Induction.			15 Hours
II	Logic and Counting Fundamentals of Logic, Propositional Logic, Logical Connectives and Truth Tables, Logic Equivalence, Tautology and Contradiction. Basics of counting, Counting Principles, Pigeonhole Principle, Permutation, Combinations.			15 Hours
III	Matrices Basics of Matrix, Types of Matrices, Operations on Matrices, Inverse of a matrix, Solution for system of linear equations, Determinant, Properties of Determinant, Cramer’s Rule, Introduction to EigenValues and EigenVectors.			15 Hours
IV	Graph Theory Graphs: Introduction, Representing Graphs, Operations on graphs, Directed Graphs Graph Isomorphism, Paths, Cycles, Euler Graph, Hamilton Graph, Planar Graphs. Trees: Introduction, Applications of Trees, Spanning Trees, Minimum Spanning Trees, Prim’s and Kruskal’s Algorithms.			15 Hours

Reference Books

- 1 "Discrete Mathematics and Its Applications" by Kenneth H. Rosen (8th Edition, 2019)
- 2 "Discrete Mathematics with Applications" by Susanna S. Epp (5th Edition, 2020)
- 3 "Linear Algebra and Its Applications" by David C. Lay, Steven R. Lay, and Judi J. McDonald (6th Edition, 2022)
- 4 "Schaum's Outline of Discrete Mathematics" by Seymour Lipschutz and Marc Lipson (4th Edition, 2021)
5. "Linear Algebra and Its Applications" by David C. Lay, Steven R. Lay, and Judi J. McDonald (6th Edition, 2022)

COURSE OUTCOME	
CO CODE	COURSE DESCRIPTION
CO1	Apply Set theory, Relations, Functions and Mathematical Induction to solve problems.
CO2	Getting familiar with counting techniques and Mathematical Logic, and applying the same to address real-life problems.
CO3	Learn different operations of a Matrix and apply them to solve real-life problems.
CO4	Analyze the use of EigenValues and EigenVectors in practical applications.
CO5	Apply graph theory concepts to solve the problems of the real world.

TEACHING PEDAGOGY

Active Learning through Practical Application, Conceptual Understanding, Problem-Based Learning, Collaborative Learning, Use of Technology, Scaffolding and Differentiation, Assessment for Learning
Formative Assessment: Regular quizzes, problem-solving sessions, and practical tasks on key concepts such as Matrices. This helps in tracking student progress and understanding.

SKILL DEVELOPMENT

Venn Diagram Applications Truth

Table Construction Permutations

and Combinations Graph Traversal

Matrix Operations

24BCA14: PROBLEM SOLVING TECHNIQUE

Course Code	24BCA14	Course Title	Problem Solving Technique	
Course Type	DSC	Contact Hours	4 Hours per Week	No of Hours: 60
Credit	4	Domain	Computer Science	
SYLLABUS				
I	Introduction		15 Hours	
	The Role of Algorithms in Computing, Algorithms as a technology, Analyzing algorithms, Designing algorithms. Fundamental Algorithms: Exchanging the values of two variables, Counting, Summation of a set of numbers, Factorial Computation, Generating of the Fibonacci sequence, Reversing the digits of an integer. Overview of C: History and importance of C, Basic structure of C program, executing a C program. Constants, Variable and Data Types: Introduction, Character Set, C Tokens, Keywords and Identifiers, Constants, Variables, Data Types, Declaration of Variables, Assigning Values to Variables, Defining Symbolic Constants			
II	C Programming		15 Hours	
	Getting Started, Arithmetic expressions. Input and Output: Standard input and output, formatted output- printf, variable length argument list, formatted input- scanf. Control Flow: Statements and Blocks, If-else, else-if, switch, loops: while loop, for loop, do while, break and continue, goto and labels. Pointers and Arrays: pointers and address, pointers and function arguments, multidimensional array, initialization of pointer arrays, command line arguments.			
III	Factoring Methods		15 Hours	
	Finding the square root of a number, the smallest Divisor of an integer, the greatest common divisor of two integers, computing the prime factor of an integer, raising a number to a large power. Array Techniques: Array order reversal, Array counting, Finding the maximum number in a set, removal of duplicates from an ordered array, partitioning an array, finding the k th smallest element, and multiplication of two matrices.			
IV	Sorting		15 Hours	
	Sorting by selection, sorting by exchange, sorting by insertion, sorting by diminishing increment, sorting by partitioning. Searching: Linear Search, Binary search. File Management in C: Introduction, Defining and opening a file, closing a file, Input/output and Error Handling on Files.			

Reference Books:

1. **"Introduction to Algorithms"** by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, 4th Edition (2022)
2. **"Programming in ANSI C"** by E. Balagurusamy, 8th Edition (2021), McGraw Hill
3. **"Let Us C"** by Yashavant Kanetkar, 17th Edition (2020), BPB Publications
4. Greg Perry and Dean Miller, "C programming Absolute Beginner's Guide", 3rd edition, Pearson Education, Inc, 2014.
5. **"Data Structures and Algorithms Made Easy in C"** by Narasimha Karumanchi 2nd Edition (2011), CareerMonk Publications.

	COURSE OUTCOME	
	CO CODE	COURSE DESCRIPTION
	CO1	To understand algorithmic strategies for enhancing problem solving proficiency
	CO2	Demonstrate problem solving tools and techniques using C.
	CO3	To analyze the given problems and use appropriate algorithms.
	CO4	To implement sorting and searching techniques to develop programs.
	CO5	Implement file handling operations in C, including opening, closing, reading, writing, and handling errors in file input/output to manage and process data efficiently.

TEACHING PEDAGOGY

Active Learning through Practical Application, Conceptual Understanding, Problem-Based Learning, Collaborative Learning, Divide-and-Conquer Approach, Think-Pair-Share, Hands-on Labs, Use of Technology, Scaffolding and Differentiation, Assessment for Learning Formative Assessment: Regular quizzes, problem-solving sessions, and practical tasks on key concepts such as Matrices. This helps in tracking student progress and understanding.

SKILL DEVELOPMENT

Code Implementation Practice

Algorithm Complexity Analysis

Debugging Exercises Algorithm

Optimization

24BCA15: COMPUTER ARCHITECTURE

Course Code	24BCA15	Course Title	Computer Architecture	
Course Type	DSC	Contact Hours	4 Hours per Week	No of Hours: 60
Credit	4	Domain	Computer Science	
SYLLABUS				
I	Number Systems 15 Hours Number Systems: Decimal, Binary, Hexadecimal, Octal Number System Conversions, Binary Arithmetic, Complements- r's complement, (r-1)'s complement, Addition and subtraction of BCD, Octal Arithmetic, Hexadecimal Arithmetic, Binary Codes, Decimal Codes, Error detecting and correcting codes, ASCII, EBCDIC, UNICODE, Digital Logic Circuits: Digital Computers, Logic Gates, Universal Gates, Boolean algebra, Map Simplification.			
II	Combinational Circuits: 15 Hours Half Adder and Full Adder, Flip-Flops- SR Flip- Flop, D Flip-Flop, J-K Flip-Flop, T Flip-Flop, Sequential Circuits- Flip-Flop input equations, State Table, State Diagram and problems. Digital Components: Integrated Circuits, Decoders-3-to-8-line decoder, NAND gate Decoder, Octal to Binary Encoder, Multiplexers- 4-to-1 line Multiplexer, Registers- 4 bit register with parallel load, Shift Registers- Bidirectional shift register with parallel load, Binary Counters-4-bit synchronous binary counter.			
III	Basic Computer Organization and Design: 15 Hours Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference Instructions, Input- Output Interrupt, Complete Computer Description, Design of Basic Computer, Design of Accumulator logic. Central Processing Unit: Introduction, General Register Organization, Stack Organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control, Reduced Instruction Set Computer (RISC), CISC Vs RISC.			
IV	Introduction to 8085 Assembly language programming: 15Hours Architecture of 8085, Pin Configuration, The 8085-programming model, Instruction classification, Instruction, data formats, and storage. overview of 8085 instruction set. Introduction to 8085 Instructions: Instruction classification of 8085 based on word length and functions, Data Transfer operations, Arithmetic operations, Logic Operations, Branch operations, Addressing modes of 8085. Overview of 8086 architecture, Pin diagram, and functional units.			

Reference Books:

1. "Digital Logic and Computer Design" by M. Morris Mano (5th Edition, 2021)
2. "Fundamentals of Digital Logic with VHDL Design" by Stephen Brown and Zvonko Vranesic (4th Edition, 2019)
3. Ramesh Gaonkar – "Microprocessor Architecture, Programming, and Applications with the 8085", 6th Edition, Penram International Publishing (India) Private Limited, 2013.
4. David A. Patterson, John L. Hennessy – "Computer Organization and Design: The Hardware/Software Interface", 5th Edition, Morgan Kaufmann, 2014.
5. A. Anand Kumar – "Fundamentals of Digital Circuits", 4th Edition, PHI Learning, 2016. Soumitra Kumar Mandal – "Microprocessors and Microcontrollers: 8085, 8086, and 8051", 2nd Edition, Tata McGraw Hill, 2013.

COURSE OUTCOME	
CO CODE	COURSE DESCRIPTION
CO1	Understand various arithmetic and logical operations on different types of numbers to design an arithmetic and logic unit.
CO2	Demonstrate design and implement sequential logic circuits using ICs.
CO3	Analyze the basics of computer organization and its design and the basic processing unit.
CO4	Implement the instruction sets and develop assembly language programming skills.
CO5	Explore modern CPU architectures and understand the principles of parallel computing to apply in high-performance computing environments.

TEACHING PEDAGOGY

Active Learning through Practical Application, Conceptual Understanding, Problem-Based Learning, Collaborative Learning, Active Learning with Peer Instruction, Visual Learning via Interactive Diagrams and Tools, Flipped Classroom Approach, Use of Technology, Scaffolding and Differentiation, Assessment for Learning, Mathematical Modeling and Simulation

SKILL DEVELOPMENT:

Analytical and Critical Thinking

Problem-Solving Skills Algorithmic

Thinking

Abstract and Formal Reasoning

Application of Mathematical Tools

Logical Reasoning and Formal Proof Writing

Teamwork and Collaboration

24BCA16: PROBLEM-SOLVING TECHNIQUE LAB

Course Code	24BCA16	Course Title	Problem Solving Technique Lab
Course Type	DSC	Contact Hours	4 Hours per Week
Credit	2	Domain	Computer Science

SYLLABUS

Write, and execute C Program for the following:

1. To exchange the values of two variables using function swap()
2. To read the numbers and find the biggest of three
3. To check whether the number is prime or not.
4. To find the summation of a set of numbers
5. To read a number, find the sum of the digits, reverse the number and check it for palindrome.
6. To convert a lowercase letter into uppercase letter using array concept
7. To read the percentage of marks and to display appropriate messages. If a percentage is 70 and above- Distinction, 60-69 – First Class, 50-59 – Second Class, 40-49 Pass, below 40 – Fail.(Demonstrate of if-else ladder)
8. To simulate a simple calculator with addition, subtraction, multiplication, division and it should display the error message for division of zero using switch case.
9. To read marks scored by n students and find the average of mark (Demonstration of single dimensional array)
10. To find the factorial of a number.
11. To generate Fibonacci series.
12. To find the greatest common divisor
13. To demonstrate string functions. (String Length, String Copy, String Concatenate, String Comparison)
14. To find the length of the string without using built-in functions.
15. To read, display and add two n x m matrices using function.
16. To read a string and to find the number of alphabets, digits, vowels, consonants, space and special characters.
17. To Swap 2 numbers using pointers
18. To demonstrate difference between structure and union
19. To generate array order reversal
20. To design the following pattern using nested loop

```

      *
    *   *
  *   *   *
*   *   *   *
*   *   *   *   *

```

COURSE OUTCOME	
CO CODE	COURSE DESCRIPTION
CO1	Demonstrate proficiency in fundamental C programming concepts for mathematical calculations and logical comparisons.
CO2	Apply array and string manipulation techniques to perform operations like searching, sorting, and modifying data structures efficiently.
CO3	Utilize advanced programming concepts such as pointers, structures, and unions to manage complex data types and memory manipulation.
CO4	Implement problem-solving strategies using loops and decision-making constructs in C to design patterns, process student data, and
CO5	Simulate real-life scenarios like calculators or grade evaluation.

TEACHING PEDAGOGY

Active Learning through Practical Application, Conceptual Understanding, Problem-Based Learning, Collaborative Learning, Use of Technology, Scaffolding and Differentiation, Assessment for Learning.

24BCA17: COMPUTER ARCHITECTURE LAB

Course Code	24BCA17	Course Title	Computer Architecture Lab
Course Type	DSC	Contact Hours	4 Hours per Week
Credit	2	Domain	Computer Science

SYLLABUS

1. Write an 8085 Program to swap two 8-bit numbers.
2. a. Write a Program to find the largest of two numbers
b. Write an 8085 Program to find the smallest of two numbers
3. Write an 8085 Program to find whether an 8-bit number is positive, negative or zero. If positive display EE, if negative display FF, if zero display DD.
4. Write an 8085 Program to check whether the 4th bit of a number is zero or one. If 4th bit is 1 display FF, if 4th bit is 0 display DD.
5. Write an 8085 Program to calculate the sum of the first ten natural numbers.
6. Write an assembly language program in 8085 microprocessors to find the sum of digits of an 8-bit number.
7. Write an 8085 Program to find the reverse of an 8-bit number
8. Write an 8085 Program to check whether 1-byte number is a palindrome or not. If it is a palindrome display FF otherwise display DD.
9. Write an 8085 Program to check whether a number is ODD or EVEN. If Even no. display DD, if odd no. display FF.
10. Write an 8085 program to count a number of ones in the given 8-bit number.
11. Write an 8085 program to find Addition & Subtraction of two 8 –bit HEX numbers.
12. Write an 8085 program to find Subtraction of two 16 –bit numbers.
13. Write an 8085 program to implement 2 out of 5 codes
14. Write an 8085 program to generate Fibonacci series
15. Write an 8085 program to find the first ten terms of odd and even numbers.
16. Write an 8085 program to find 4-Digit BCD addition.
17. Write an 8085 program to find Multiplication of 2-digit BCD numbers.

18. Write an 8085 program to find division of two 8-bit numbers.
19. WRITE AN 8086 PROGRAM TO MULTIPLY TWO NUMBERS
20. WRITE AN 8086 PROGRAM TO REVERSE A STRING

COURSE OUTCOME	
CO CODE	COURSE DESCRIPTION
CO1	Write and debug 8085 assembly programs to perform basic arithmetic operations, including addition, subtraction, and multiplication of 8-bit and 16-bit numbers.
CO2	Develop 8085 programs to manipulate and analyze 8-bit numbers by implementing functions such as swapping, reversing, and checking for palindromes.
CO3	Create 8085 programs to handle and display results based on bitwise operations, including checking specific bits and determining the parity of a number.
CO4	Implement 8085 programs for more advanced number processing tasks, such as generating Fibonacci series, summing digits of numbers, and performing BCD arithmetic.
CO5	Design and execute 8085 assembly programs to work with codes and data conversions, such as 2 out of 5 codes and finding the sum of first ten natural numbers.

TEACHING PEDAGOGY

Active Learning through Practical Application, Conceptual Understanding, Problem-Based Learning, Collaborative Learning, Use of Technology, Scaffolding and Differentiation, Assessment for Learning.

24BCA18: OFFICE AUTOMATION TOOLS LAB

Course Code	24BCA18	Course Title	Office Automation Tools Lab
Course Type	DSC	Contact Hours	4 Hours per Week
Credit	2	Domain	Computer Science

SYLLABUS

1. Document Formatting
2. Resume Preparation
3. Mail Merge
4. Working with Tables
5. Data Entry and Formatting
6. Formula and Functions
7. Sorting, Filtering, and Charts
8. Pivot Table
9. Creating a Basic Presentation
10. Advanced Presentation Features
11. Custom Animation and Timings
12. Creating and Managing a Database
13. Querying the Database
14. Creating Forms and Reports
15. Relational Database Concepts
16. Email and Calendar
17. File and Folder Operations

COURSE OUTCOME	
CO CODE	COURSE DESCRIPTION
CO1	Write and debug 8085 assembly programs to perform basic arithmetic operations, including addition, subtraction, and multiplication of 8-bit and 16- bit numbers.
CO2	Develop 8085 programs to manipulate and analyze 8-bit numbers by implementing functions such as swapping, reversing, and checking for palindromes.
CO3	Create 8085 programs to handle and display results based on bitwise operations, including checking specific bits and determining the parity of a number.
CO4	Implement 8085 programs for more advanced number processing tasks, such as generating Fibonacci series, summing digits of numbers, and performing BCD arithmetic.
CO5	Design and execute 8085 assembly programs to work with codes and data conversions, such as 2 out of 5 codes and finding the sum of first ten natural numbers.

TEACHING PEDAGOGY

Active Learning through Practical Application, Conceptual Understanding, Problem-Based Learning, Collaborative Learning, Use of Technology, Scaffolding and Differentiation, Assessment for Learning.

CURRICULUM ENHANCEMENT

Certificate Course to be included from Spoken Tutorial IIT Bombay

SEMESTER II

24BCA23: DATA STRUCTURES

Course Code	24BCA23	Course Title	Data Structures
Course Type	DSC	Contact Hours	4 Hours per Week No of Hours: 60
Credit	4	Domain	Computer Science
SYLLABUS			
I	Introduction and Overview 15 Hours Definition, Elementary data organization, Data Structures, data Structures operations, Abstract data types, algorithms complexity, time-space trade-off. Preliminaries: Mathematical notations and functions, Algorithmic notations, control structures, Complexity of algorithms, asymptotic notations for complexity of algorithms. Introduction to Strings, Storing String, Character Data Types, String Operations, word processing, Introduction to pattern matching algorithms.		
II	Arrays and Linked List 15 Hours Definition, Linear arrays, arrays as ADT, Representation of Linear Arrays in Memory, Traversing Linear arrays, Inserting and deleting, multi-dimensional arrays, Matrices and Sparse matrices, searching and sorting techniques using array. Linked list: Definition, Representation of Singly Linked List in memory, Traversing a Singly linked list, Searching in a Singly linked list, Memory allocation, Garbage collection, Insertion into a singly linked list, Deletion from a singly linked list; Doubly linked list, Header linked list, Circular linked list.		
III	Stacks and Queues: 15 Hours Definition, Array representation of stacks, Linked representation of stacks, Stack as ADT, Arithmetic Expressions: Polish Notation, Conversion of infix expression to postfix expression, Evaluation of Postfix expression, Application of Stacks, Recursion, Towers of Hanoi, Implementation of recursive procedures by stack. Queues: Definition, Array representation of queue, Linked list representation of queues. Types of queue: Simple queue, Circular queue, Double-ended queue, Priority queue, Operations on Queues, Applications of queues. Searching and Sorting: Linear and Binary search, bubble sort, selection sort, insertion sort and heap sort.		
IV	Binary Trees: 15 Hours Definitions, Tree Search, Traversal of Binary Tree, Tree Sort, Building a Binary Search Tree, Height Balance: AVL Trees, Contiguous Representation of Binary Trees: Heaps, Red Black Tree: Insertion and Deletion, External Searching: B-Trees, Applications of Trees. Graphs: Mathematical Background, Computer Representation, Graph Traversal.		

	Hashing: Hash Table ADT, understanding Hashing, Components of Hashing, Hash Table, Hash Function, Hashing Techniques, collisions, collision resolution techniques.
--	--

COURSE OUTCOME	
CO CODE	COURSE DESCRIPTION
CO1	Understand basic concepts of data structures.
CO2	Analyzing and exploring various ways of storing data using Array and Linked list.
CO3	Demonstrate stack and queue data structures and their applications
CO4	Analyze and implement various nonlinear data structures.
CO5	Design and analyze searching (linear and binary) and sorting techniques

Reference Books:

1. Seymour Lipschutz, “Data Structures with C”, Schaum’s Outlines, Tata Mc Graw Hill, 2011.
2. Robert Kruse, C.L.Tondo, Bruce Leung, Shashi Mogalla, “Data Structures and Program Design using C”, Pearson Education, 2010
3. Mark Allen Weiss, “Data Structures and Algorithm Analysis in C”, Second Edition, Pearson Education, 2013
4. Forouzan, “A Structured Programming Approach using C”, 2nd Edition, Cengage Learning India, 2008.

TEACHING PEDAGOGY

Active Learning through Game-Based Learning, Practical Application, Simulations, Conceptual Understanding, Problem-Based Learning, Collaborative Learning, Use of Technology, Scaffolding and Differentiation, Assessment for Learning. Formative Assessment: Regular quizzes, problem-solving sessions, and practical tasks on key concepts such as Matrices. This helps in tracking student progress and understanding.

SKILL DEVELOPMENT

Hands-On

Analysis Group

Activities

Pattern Matching Practice

Error Handling

24BCA24: OBJECT ORIENTED PROGRAMMING USING JAVA

Course Code	24BCA24	Course Title	Object Oriented Programming Using Java	
Course Type	DSC	Contact Hours	4 Hours per Week	No of Hours: 60
Credit	4	Domain	Computer Science	
SYLLABUS				
I	Introduction		15 Hours	
	Basics of object-oriented programming, comparison of procedure-oriented and object- oriented programming paradigms; Difference between C and Java Programming languages; Features of Java; Objects and classes in Java, Structure of a Java program; Data Types, variables and operators in java; Control structures- Branching and looping; Methods & Constructors in java; Java Development Kit (JDK); Built-in classes in Java; Math, Character, String, String Buffer and Scanner; Wrapper classes; The abstract, static and final classes; Casting objects; The instance of operator; Usage of this keyword; Arrays in Java.			
II	Inheritance:		15 Hours	
	Super and subclasses; visibility modifiers; Types of Inheritance- single, multilevel, hierarchical and hybrid inheritance; the interface concept in Java, Polymorphism: Compile time and runtime polymorphism – Method overloading and method overriding. Package: Types of packages; the util, awt and swing packages; Creating and importing user-defined packages. I/O programming: Standard I/O streams in Java; Types of streams – Based on the type of Operations and the type of file.			
III	Event Handling:		15 Hours	
	Major events in Java; Two Event Handling mechanisms- Event classes and Event Listener Interfaces; Mouse and keyboard events; GUI: Panels; Frames; Layout managers – Flow, border and grid layouts; Buttons; Checkboxes; Radio buttons; Labels; Text fields; Text areas; Combo boxes; Scroll bars; Sliders; Menu, Dialog boxes. Applet programming: Comparison of applets and applications; Applet life cycle; Developing and running applets. String handling: String construction, string length, special string operations, character extraction, string comparison, modifying string and string buffers.			
IV	Exception Handling:		15 Hours	
	Types of Java exception – checked and unchecked exceptions; Usage of try- catch-finally blocks. Multithreading: comparison of multithreading and multitasking; Life cycle of a thread; two ways of creating thread – by extending the Thread class			

	and by implementing the Runnable Interface, Thread synchronization. Advanced concepts: Collections in Java; Introduction to JavaBeans and Java security manager, Importance of generic programming in java with examples. JDBC and ODBC Connections. Simple project in java with JDBC and ODBC.
--	---

Reference Books

1. E. Balagurusamy, Programming with JAVA, McGraw Hill, New Delhi, 2007
2. Herbert Schildt, Java A Beginner's Guide – Create, Compile, and Run Java Programs Today, Sixth Edition, Oracle Press, 2014
3. Ken Arnold, James Gosling, "The Java Programming Language, Fourth Edition, Addison Wisely, 2005
4. R Herbert Schildt, 'The Complete Reference Java, 7th Edition, McGraw Hill, 2007

COURSE OUTCOME	
CO CODE	COURSE DESCRIPTION
CO1	Understand object-oriented programming concepts
CO2	Demonstrate the important feature of Object-oriented programming
CO3	Examine event handling, String handling, and exception handling concepts
CO4	Implement concepts to solve real-world problems
CO5	Design and implement a simple project in Java that integrates JDBC or ODBC to interact with a database

TEACHING PEDAGOGY

Active Learning through Practical Application, Conceptual Understanding, Problem-Based Learning, Collaborative Learning, Use of Technology, Scenario-Based Learning, Assessment for Learning Formative Assessment: Regular quizzes, problem-solving sessions, and practical tasks on key concepts such as Matrices. This helps in tracking student progress and understanding.

SKILL DEVELOPMENT

Identify Object-Oriented

Features Incremental

Programming Exception

Handling Tasks Thread

24BCA25: OPERATING SYSTEMS

Course Code	24BCA25	Course Title	Operating Systems	
Course Type	DSC	Contact Hours	4 Hours per Week	No of Hours: 60
Credit	4	Domain	Computer Science	
SYLLABUS				
I	Introduction: 15 Hours Computer System Organization, Architecture, Structure, Operations, Process Management, Memory Management, Storage Management. Operating System Structures: Services, System Calls, Types, Operating System Structure, System Boot. Processes: Process Concept, Scheduling, Operations, Inter-process Communication. Multithreaded Programming: Multithreading Models.			
II	Process Synchronization: 15 Hours The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors, Synchronization Examples. Process Scheduling: Criteria, Scheduling Algorithms, Multi-Processor Scheduling, Real-time CPU Scheduling. Deadlocks: System model, Characterization, Methods for handling deadlocks, Deadlock Prevention, Avoidance, Detection and Recovery from deadlock.			
III	Memory Management Strategies: 15 Hours Background, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table. Virtual Memory Management: Demand Paging; Copy-on-Write, Page Replacement; Allocation of Frames, File System: File Concept, Access Methods, Directory and Disk Structure, Protection. File- system Implementation: Structure, File- System and Directory Implementation, Allocation Methods, Free Space Management. Mass-Storage Structure: Overview, Disk Scheduling, Disk Management. System Protection: Goals - Domain - Access matrix. System Security: The Security Problem - Threats –			

	Encryption- User Authentication.
IV	Introduction to Linux Programming: 15 Hours Linux system Architecture, Linux Command format, Linux Internal and External Commands, Directory Commands, File related commands, Disk related commands, General Utilities.

Reference Books

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne; "Operating Systems Concepts", 9th Edition, 2016 India, Wiley.
2. William Stallings, "Operating Systems and Design Principles", Pearson, 5th Edition, 2018
3. D M Dhamdhere : Operating Systems - A concept Based Approach, 3rd Edition, Tata McGraw-Hill, 2017.
4. Sumitabha Das: "UNIX Concepts and Applications", 4th Edition, Tata McGraw Hill, 2017.
5. M G Venkateshmurthy, Introduction to UNIX and Shell Programming", Pearson Edition Asia, 2005.

COURSE OUTCOME	
CO CODE	COURSE DESCRIPTION
CO1	To analyze the memory management and its allocation policies
CO2	To understand synchronization and deadlock conditions and their possible solutions
CO3	To discuss the storage management policies with respect to different storage
CO4	To evaluate the concept of the operating system with respect to Linux architecture and commands
CO5	To understand and apply system protection mechanisms, including access control using the access matrix model.

TEACHING PEDAGOGY

Active Learning through Practical Application, Conceptual Understanding, Problem-Based Learning, Collaborative Learning, Use of Technology, Scaffolding and Differentiation, Assessment for Learning Formative Assessment: Regular quizzes, problem-solving sessions, and practical tasks on key concepts such as Matrices. This helps in tracking student progress and understanding.

SKILL DEVELOPMENT

System Design Simulation

Command-Line Exercises

System Utilities

Disk Scheduling Simulation

Paging & Segmentation

24BCA26: DATA STRUCTURES LAB

Course Code	24BCA26	Course Title	Data Structures Lab
Course Type	DSC	Contact Hours	4 Hours per Week
Credit	2	Domain	Computer Science

SYLLABUS

1. Write a program to search for an element in an array using binary and linear search.
2. Write a program to sort a list of n numbers using Bubble Sort algorithms.
3. Perform the Insertion and Selection Sort on the input {75,8,1,16,48,3,7,0} and display the output in descending order.
4. Write a program to insert the elements {61,16,8,27} into a singly linked list and delete 8,61,27 from the list. Display your list after each insertion and deletion.
5. Write a program to insert the elements {45, 34, 10, 63,3} into a linear queue and delete three elements from the list. Display your list after each insertion and deletion.
6. Write a program to simulate the working of a Circular queue using an array.
7. Write a program to insert the elements {61,16,8,27} into an ordered singly linked list and delete 8,61,27 from the list. Display your list after each insertion and deletion.
8. Write a program for the Tower of Hanoi problem using recursion.
9. Write a recursive program to find GCD of 3 numbers.
10. Write a program to demonstrate working of stack using linked list.
11. Write a program to convert an infix expression $x^y/(5*z)+2$ to its postfix expression
12. Write a program to evaluate a postfix expression $5\ 3+8\ 2\ -\ *$.
13. Write a program to create a binary tree with the elements {18,15,40,50,30,17,41} after creation insert 45 and 19 into the tree and delete 15,17 and 41 from the tree. Display the tree on each insertion and deletion operation.
14. Write a program to create a binary search tree with the elements {2,5,1,3,9,0,6} and perform inorder, preorder and post order traversal.
15. Write a program to Sort the following elements using heap sort {9,16,32,8,4,1,5,8,0}.
16. Given $S1=\{\text{"Flowers"}\}$; $S2=\{\text{"are beautiful"}\}$ I. Find the length of S1 II. Concatenate S1 and S2 III. Extract the substring "low" from S1 IV. Find "are" in S2 and replace it with "is.
17. Write a program to implement an adjacency matrix of a graph.
18. Write a program to insert/retrieve an entry into hash/ from a hash table with open

addressing using linear probing..

COURSE OUTCOME	
CO CODE	COURSE DESCRIPTION
CO1	Analyze and implement basic searching and sorting algorithms using arrays.
CO2	Develop and manipulate linear and non-linear data structures like linked lists, stacks, queues, and trees.
CO3	Solve problems using recursion and understand its application in algorithmic solutions.
CO4	Apply stack and queue operations in real-world applications like expression conversion and evaluation.
CO5	Implement and analyze advanced data structures such as heaps, hash tables, and graphs.

TEACHING PEDAGOGY

Active Learning through Practical Application, Conceptual Understanding, Problem-Based Learning, Collaborative Learning, Use of Technology, Scaffolding and Differentiation, Assessment for Learning

24BCA27: OBJECT-ORIENTED PROGRAMMING LAB

Course Code	24BCA27	Course Title	Object-Oriented Programming Lab
Course Type	DSC	Contact Hours	4 Hours per Week
Credit	2	Domain	Computer Science

SYLLABUS

1. Java program to display “Hello World” and display the size of all the data types.
2. Java program to implement the usage of static, local and global variables.
3. Java program to implement string operations string length, string concatenate, substring
4. Java program to find the maximum of three numbers
5. Java program to check whether the number is odd or even.
6. Java program to implement default and parameterized constructors.
7. Java program to implement an array of objects.
8. Java program to implement Single Inheritance
9. Java program to implement Multiple Inheritance using Interface
10. Java program to implement an applet
11. Java program to demonstrate a division by zero exception
12. Java program to add two integers and two float numbers. When no arguments are supplied, give a default value to calculate the sum. Use method overloading.
13. Java program that demonstrates run-time polymorphism.
14. Java program to catch negative array size Exception. This exception is caused when the array is initialized to negative values.
15. Java program to handle null pointer exception and use the “finally” method to display a message to the user.
16. Java program to import user-defined packages
17. Java program to check whether a number is palindrome or not
18. Java program to find the factorial of a list of numbers reading input as command line argument.
19. Java program to display all prime numbers between two limits.
20. Java program to create a thread using Runnable Interface.

COURSE OUTCOME	
CO CODE	COURSE DESCRIPTION
CO1	Understand and apply basic Java syntax, data types, and control structures.
CO2	Implement object-oriented programming concepts such as inheritance, polymorphism, and encapsulation in Java.
CO3	Demonstrate exception handling and error management using built-in and custom exceptions.
CO4	Work with Java string manipulation and methods to perform various string operations.
CO5	Implement multi-threading and applet-based applications in Java.

TEACHING PEDAGOGY

Active Learning through Practical Application, Conceptual Understanding, Problem-Based Learning, Collaborative Learning, Use of Technology, Scaffolding and Differentiation, Assessment for Learning

24BCA28: LINUX AND SHELL PROGRAMMING LAB

Course Code	24BCA28	Course Title	Linux and Shell Programming Lab
Course Type	DSC	Contact Hours	4 Hours per Week
Credit	2	Domain	Computer Science
SYLLABUS			
1. Write a shell script to print all the prime numbers between M to N ($M < N$). 2. Write a shell script to reverse a given number and check whether it is a palindrome. 3. Write a shell script to find the sum of digits of a given number using loops and without using loops. 4. Write a shell script to implement 10 Linux commands using case. 5. Write a Shell script that displays a list of all the files in the current directory to which the user 6. has read, write and execute permissions? 7. Write a shell script to copy a file within the current directory 8. Write a shell script to copy file between two directories 9. Write a Shell script to create two data files and compare them to display unique and common entries. 10. Write a shell script to count the number of vowels in a string. 11. Write a shell script to convert uppercase characters to lowercase and vice versa. 12. Write a shell script to accept a word and perform pattern matching in a given file. 13. Write a shell script to find the factorial of a number 14. Write a Menu-driven program to demonstrate the zombie process and orphan process.			

COURSE OUTCOME	
CO CODE	COURSE DESCRIPTION
CO1	Develop shell scripts to perform basic numerical operations and decision-making.
CO2	Demonstrate file handling and directory management using shell scripting.
CO3	Implement string manipulation and pattern matching in shell scripts.
CO4	Apply conditional execution and control structures in scripts to execute system commands.
CO5	Utilize shell scripting to automate comparisons and manage data files.

TEACHING PEDAGOGY

Active Learning through Practical Application, Conceptual Understanding, Problem-Based Learning, Collaborative Learning, Use of Technology, Scaffolding and Differentiation, Assessment for Learning.

CURRICULUM ENHANCEMENT

Internship is to be added.