UNIVERSITY OF CALCUTTA

Notification No. CSR/79/2026

It is notified for information of all concerned that in terms of the provisions of Section 54 of the Calcutta University Act, 1979, (as amended), and, in the exercise of his powers under 9(6) of the said Act, the Vice-Chancellor has, by an order dated 15.07.2026, approved the Syllabus for semester-7 of 4-Year Honours & Honours with Research Courses of Studies in **Computer Science under CCF, 2022.**

The above shall take effect from the Odd Semester Examinations, 2026, and onwards.

22/07/2026

SENATE HOUSE
Kolkata-700073
22.07.2026

Prof.(Dr.) Debasis Das
Registrar

University of Calcutta

B.Sc. (Honours and Honours with Research), 4- Years degree program in Computer Science under Curriculum & Credit Framework (CCF).

Semester – VII

Semester - 7

Paper Code	Theory Paper	Credit	Contact hours	Practical/Tutorial Paper	Credit	Contact hours
DSCC-16	Compiler Design	03	45	Tutorial	1	30
DSCC-17	AI & Machine Learning	03	45	AI &Machine Learning Lab using Python	1	30
DSCC-18	Computer Graphics	03	45	Computer Graphics Lab	1	30
DSCC-19	Embedded Systems and application to IoT	03	45	Project	1	30
DSCC-20	Advanced DBMS	03	45	Tutorial	1	30

DSCC-16-Th: Theory: Compiler Design
Credits - 03, Contact hours – 45, Marks-75

Introduction: Phases of compilation and overview.	**2 hrs**
Lexical Analysis (scanner): Regular language, finite automata, regular expression, from regular expression to finite automata, scanner generator (lex, flex).	**6 hrs**
Syntax Analysis (Parser): Context-free language and grammar, push-down automata, LL(1) grammar and top-down parsing, operator grammar, LR(O), SLR(1), LR(1), LALR(1) grammars and bottom-up parsing, ambiguity and LR parsing, LALR(1) parser generator (yacc, bison)	**10 hrs**
Semantic Analysis: Attribute grammar, syntax directed definition, evaluation and flow of attribute in a syntax tree.	**4 hrs**
Symbol Table: Its structure, symbol attributes and management.	**3 hrs**
Run-time environment: Procedure activation, parameter passing, value return, memory allocation, and scope.	**4 hrs**
Intermediate Code Generation: Translation of different language features, different types of intermediate forms.	**4 hrs**
Code Improvement (optimization): Analysis: control-flow, data-flow dependence etc.; Code improvement local optimization, global optimization, loop optimization, peep-hole optimization etc. Architecture dependent code improvement: instruction scheduling (for pipeline), loop optimization (for cache memory) etc.	**6 hrs**
Register allocation and target code generation.	**3 hrs**
Advanced topics: Type systems, data abstraction, compilation of object oriented features and non-imperative programming languages.	**3 hrs**

DSCC-16-TU: Tutorial
Credits - 01, Contact hours – 30, Marks-25

References

1. Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, Compilers: Principles, Techniques and Tools, Addison-Wesley.
2. Michael L. Scott, Programming Language Pragmatics, Elsevier.
3. Andrew W. Appel, Modern Compiler Implementation in C/Java, Cambridge University Press.
4. Keith D. Cooper and Linda Torczon, Engineering a Compiler, Elsevier.
5. Allen I. Holob, Compiler Design in C, Prentice-Hall.
6. Steven S. Muchnik, Advanced Compiler Design and Implementation, Elsevier.
7. Randy Allen and Ken Kennedy, Optimizing Compilers for Modern Architectures, Elsevier.

DSCC-17-Th: Theory: AI & Machine Learning
Credits - 03, Contact hours – 45, Marks-75

Introduction to Artificial Intelligence Introduction to Artificial Intelligence; evolution of Artificial Intelligence; overview of Classical AI / Symbolic AI; basic concept of intelligent agents; introductory idea of problem-solving in AI; overview of state space representation; basic search-based problem solving; elementary idea of uninformed search with examples of breadth-first search and depth-first search; introductory concept of informed and heuristic search; basic idea of knowledge representation; rule-based reasoning; overview of production rules; introduction to expert systems and their applications; limitations of Classical AI	**7 hrs**
Introduction to Machine Learning, Data Pre-processing and Model Evaluation Introduction to Machine Learning and Data Science; need and applications of Machine Learning; types of Machine Learning: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning, data collection and representation; data cleaning; handling missing values; encoding categorical data; feature scaling and normalization; train-test split; cross-validation; confusion matrix; accuracy, precision, recall, F1-score; mean absolute error, mean squared error, bias and variance; overfitting and underfitting	**8 hrs**
Supervised Learning – Regression Concept of regression; simple linear regression; multiple linear regression; polynomial regression; cost function; gradient descent; model fitting and prediction; evaluation of regression models; bias-variance trade-off.	**6hrs**
Supervised Learning – Classification Concept of classification; logistic regression; k-nearest neighbours; decision tree; random forest; naive Bayes classifier; support vector machine; comparison of classification algorithms, ensemble technique introduction, bagging, boosting, random forest	**12 hrs**
Unsupervised Learning Concept of unsupervised learning; clustering and its applications; k-means clustering; hierarchical clustering; dimensionality reduction; principal component analysis; evaluation of clustering results.	**6 hrs**
Introduction to Neural Networks and Advanced Topics Introduction to artificial neural networks; perceptron; activation functions; feed-forward neural network; basics of backpropagation; introduction to deep learning; overview of reinforcement learning; ethical issues in Machine Learning; bias, fairness and explainability.	**6 hrs**

DSCC-17-P, Practical: AI & Machine Learning Lab using Python
Credits - 01, Contact hours – 30, Marks-25

Lab Assignments: (These are illustrative examples; additional items may be incorporated to support practice and examination requirements).

1. Write a Python program to load the Iris dataset, split it into training and testing sets, and apply cross-validation using any suitable classifier.
   Dataset: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html.

2. Write a Python program to implement simple linear regression using one feature from the Diabetes dataset and evaluate it using MAE, MSE and RMSE.
   Dataset: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_diabetes.html.

3. Write a Python program to implement multiple linear regression using the California Housing dataset and evaluate the predicted house values — Dataset: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html.

4. Write a Python program to implement polynomial regression on a suitable regression dataset and compare it with simple linear regression — Reference: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html.

5. Write a Python program to implement logistic regression using the Breast Cancer Wisconsin dataset and evaluate the model using confusion matrix, precision, recall and F1-score — Dataset: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html.

6. Write a Python program to implement k-nearest neighbours classification using the Iris dataset and compare accuracy for different values of k — Dataset: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html.

7. Write a Python program to implement Naive Bayes classification using the 20 Newsgroups text dataset — Dataset: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.html.

8. Write a Python program to implement decision tree classification using the Titanic dataset and evaluate survival prediction — Dataset: https://www.kaggle.com/c/titanic/data.

9. Write a Python program to implement random forest classification using the Breast Cancer Wisconsin dataset and identify important features — Dataset: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html.

10. Write a Python program to implement support vector machine classification using the Digits dataset and evaluate handwritten digit prediction — Dataset: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html.

11. Write a Python program to train two classification models on the same dataset and compare them using accuracy, precision, recall and F1-score — Dataset: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html.

12. Write a Python program to compare Logistic Regression, Decision Tree, Random Forest and SVM on the same classification dataset — Dataset: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html.

13. Write a Python program to implement k-means clustering using the Mall Customer Segmentation dataset and visualize the customer clusters — Dataset: https://www.kaggle.com/datasets/vjchoudhary7/customer-segmentation-tutorial-in-python

Suggested Text Books:

1. *Machine Learning*, Tom M. Mitchell, McGraw Hill.
2. *Introduction to Machine Learning*, Ethem Alpaydin, MIT Press.

3. *Machine Learning*, Saikat Dutt, Subramanian Chandramouli and Amit Kumar Das, Pearson Education India.
4. *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow*, Aurélien Géron, O'Reilly.
5. *Pattern Recognition and Machine Learning*, Christopher M. Bishop, Springer.
6. *The Elements of Statistical Learning*, Trevor Hastie, Robert Tibshirani and Jerome Friedman, Springer.
7. *Artificial Intelligence: A Modern Approach*, Stuart Russell and Peter Norvig, Pearson.

DSCC-18-Th: Theory: Computer Graphics
Credits - 03, Contact hours – 45, Marks-75

Introduction Basic concepts of Graphics Devices– Monochrome and Color Monitor displaying technique only and Printing technique of Printer device, Physical and logical units of graphics devices – Pixel and its different properties, Basic idea for image or picture formation using pixels – Raster Scan and Vector Scan.	**3 hrs**
2D & 3D Transformations Geometric Transformations operations - Translation, Rotation, Scaling, Reflection, Shearing and Inverse of these operations, Homogeneous coordinate system representation, matrix representation of Coordinate Transformations operations, matrix representation of Composite Transformations Operations – Basic ideas and matrix representations by matrix concatenation for a particular operation.	**12hrs**
Basic geometrical shapes formation algorithms Concepts of Co-ordinate System, Line Segment, Circle and arc segment, elliptic segment and its formation DDA, Bresenham's and Midpoint scan conversion algorithms. Basic Operations on Images	**12 hrs**
Two-dimensional clipping Point Clipping, Line Clipping – Region coding, Cohen-Sutherland algorithm,Polygon clipping – Sutherland Hodgeman algorithm	**10 hrs**
Video Games & Animation Fundamentals of video game graphics, Computer animation principles.	**8hrs**

DSCC-18-P: Practical: Computer Graphics Lab using Python and OpenCV
Credits - 01, Contact hours – 30, Marks-25

Lab Assignments: (These are illustrative examples; additional items may be incorporated to support practice and examination requirements).

1. Implement Bresenham's algorithm and compare with OpenCV's cv2.line().
2. Circle Drawing – Generate a circle using the midpoint algorithm and verify with cv2.circle().
3. 2D Rotation – Rotate a triangle by 45° using transformation matrices.
4. 2D Scaling – Scale a polygon by a factor of 2× and visualize.
5. All 2D and 3D transformations, composite transformations.
6. Line Clipping – Implement Cohen-Sutherland line clipping for given segments.
7. 3D Cube Transformation – Construct a cube in Matplotlib and apply translation along z-axis.

8. Bezier Curve Generation – Generate a cubic Bezier curve with four control points.
9. Polygon clipping algorithm – Implement Sutherland Hodgeman polygon clipping algorithm
10. Sprite Animation – Animate a moving character across the screen using OpenCV.
11. Keyframe Animation – Interpolate between two positions of an object for smooth motion.
12. Simulate a bouncing ball with gravity and collision detection.

Suggested Text books

- Computer graphics : algorithms and implementations, Mukherjee, D. P, Jana, Debasish, PHI.
- Computer Graphics by Hearn & Baker, Prentice Hall.
- Computer Graphics Principles and Practice by James D. Foley, Pearson India.
- Introduction to Computer Graphics and Multimedia, 2nd Edition by Anirban Mukhopadhyay, Arup Chattopadhyay, Vikas Publications.
- Computer Graphics: With An Introduction To Multimedia by Rajiv Chopra, S Chand.

DSCC-19-Th: Theory: Embedded Systems and application to IoT
Credits - 03, Contact hours – 45, Marks-75

Embedded Systems Introduction to Embedded Systems, Characteristics, Applications, System-on-chip (SOC) vs Microcontrollers vs Microprocessors, Selection criteria for IoT applications, Silicon Cores: ARM based Architecture,Memory Architecture, Input/Output Processing: GPIO, DMA, Real time event handling, NVIC.	**6 hrs**
Real-Time Operating Systems (RTOS) **RTOS essentials:**Concepts of determinism, hard vs. soft real-time system constraints, multitasking execution, and context switching over a single core, **Kernel foundations:**ask creation, task states (Running, Ready, Blocked, Suspended), static/dynamic priorities, and scheduling policies (Rate Monotonic, Earliest Deadline First), **Inter-Task Communication:**Synchronizing shared resources using Semaphores and Mutexes; avoiding priority inversion; message queues and event flags, **Industry Frameworks (Qualitative understanding only):**Architectural overview and configuration setups for **FreeRTOS** and **Zephyr RTOS**.	**7 hrs**
Introduction to IoT Definition and Concepts of IoT, IoT Architecture, Components of IoT, IoT Ecosystem, Applications of IoT in various domains.	**4 hrs**
Sensors and Actuators **Analog Sensors & Interfacing:**Operational behaviour of Light Dependent Resistors (LDR), thermistors, soil moisture probes, and MQ-series gas sensors, thermostat. **Signal conditioning:** Operational Amplifiers (Op-Amps). **Digital Sensor Modules:** Data collection and timing requirements for DHT11/DHT22 (Temperature & Humidity), HC-SR04 (Ultrasonic distance), and MPU6050 (I2C-based Accelerometer/Gyroscope). **Display:** LED Matrix, LCD and OLED display. **Actuation & Data Conversion:** Driving high-voltage loads using electromagnetic Relays, controlling DC, Servo, and Stepper motors via Pulse Width Modulation (PWM),principles of Analog-to-Digital Converters (ADC) and Digital-to-Analog Converters (DAC).	**5 hrs**

IoT Networking Layers **Local wireless topologies;**Wi-Fi 6E/7, BLE Mesh,**Long-range wide-area networking:**LoRa WAN, NB-IoT, **Internet application layers**: MQTT/MQTT-SN publish-subscribe model, CoAP, HTTP/REST, and modern smart-home interoperability standards (Matter over Thread).	**5 hrs**
IoT Hardware Platforms Arduino, Raspberry Pi, NodeMCU, ESP8266, Interfacing with sensors, Hands-on programming basics, **Peripheral Serial Protocol:**UART, I2C, SPI.	**5 hrs**
Cloud and Data Analytics IoT Cloud Platforms, Data Storage, Data Visualization, Edge vs Cloud computing.	**7 hrs**
Security and Applications IoT Security Issues, Encryption Basics, Privacy Concerns, Case Studies: Smart Homes,Healthcare, Agriculture, Smart Cities.	**6 hrs**

DSCC-19-P: Project
Credits - 01, Contact hours – 30, Marks-25.

Hardware Platforms / Components required for projects.

Hardware development board		
1	Microcontroller Development board	Arduino Uno/ Arduino Mega
2	Microcontroller Development board	ESP 32/NodeMCU ESP8266
3	Single board computer	Raspberry Pi – 3 (1GB)/ 4 (1GB) or higher version
Sensors		
1	Temperature Sensor	DHT- 11/22
2	Pressure Sensor	BMP 180/280
3	Ultrasonic Sensor	HC-SR04
4	GPS Sensor	Neo-6M/8M
5	Orientation / Gyro sensor	MPU6050
6	Air Quality Sensor	MQ2 and MQ135
7	Soil Moisture sensor	YL-69
8	Current sensor	ACS712
9	Rain Sensor	YL-83
10	PIR Sensor	HC-SR501
11	RF id reader	RC522
12	RF id Card	1K Mifare RFid Card 13.65MHz
13	Wifi Module	ATWINC150
14	Camera Module-ESP32 CAM	OV2640
15	Bluetooth Module	HC-05/06 or HM-10
16	Buzzer	
17	LCD Module, 16 x 2	HD44780 controller
18	OLED	SH1106
19	Servo Motor	SG90 Micro Servo
20	Stepper Motor	28BYJ-48
21	LDR Sensor	Module based on LM393

22	IR proximity Sensor module	

Embedded Systems and application to IoT projects(These are illustrative examples of projects that may be undertaken, with a stronger emphasis on Embedded Systems and their practical applications within the domain of IoT and related fields.)

1. LED Blinking using Embedded System
Design a simple embedded system to blink an LED at regular intervals. Explain the working process and timing control used in the program.

2. Automatic Fan Controller
Develop a system where a fan automatically turns ON when temperature increases above a fixed value. Explain the role of sensor and controller.

3. Calculator using Microcontroller
Create a basic calculator system using a microcontroller for addition and subtraction operations. Describe input and output processing.

4. Smart Traffic Light Controller using Embedded System
Design a smart traffic light controller for a four-road junction using an embedded system. The system should dynamically adjust timing based on traffic density. Explain the architecture, required hardware, and control logic.

5. Real-Time Embedded System for Fire Detection
Design a real-time embedded fire alarm system capable of detecting smoke and temperature rise. Discuss interrupt handling, response time, and system reliability.

IoT-Arduino-Based Design Problems

6. Arduino Smart Traffic Light System
Design a traffic light controller using Arduino and LEDs. The system should automatically change signals after fixed intervals and include pedestrian crossing support.

7. Arduino Temperature Monitoring System
Develop a temperature monitoring system using Arduino and a temperature sensor. Display the temperature value on an LCD screen and activate a buzzer when temperature exceeds a limit.

8. Arduino Smart Irrigation System
Design an automatic irrigation system using Arduino and soil moisture sensors. The water pump should automatically turn ON or OFF depending on soil condition.

9. Arduino Home Security Alarm
Create a simple home security system using Arduino, PIR sensor, and buzzer. The system should detect movement and trigger an alarm.

10. Arduino Automatic Street Light
Design an automatic street lighting system using Arduino and LDR sensor. The lights should turn ON during night and OFF during daytime.

11. Arduino Smart Parking Counter
Develop a parking slot counter using Arduino and IR sensors. The system should display available parking spaces using LEDs or LCD.

12. Arduino Gas Leakage Detection System
Design a gas leakage monitoring system using Arduino and gas sensors. The system should activate an alarm when gas concentration crosses a threshold value.

13. Arduino Digital Door Lock
Create a password-based door locking system using Arduino and keypad module. The lock should open only for correct passwords.

ESP8266 /NodeMCU/ESP32-Based Design Problems

14. NodeMCU Wi-Fi Weather Monitoring System
Design a weather monitoring system using NodeMCU ESP8266 /ESP 32 and sensors. Sensor data should be uploaded to a cloud platform through Wi-Fi.

15. ESP8266 Smart Home Automation
Develop a smart home system using ESP8266 where appliances like fan and light can be controlled from a mobile phone.

16. NodeMCU Moisture Detection System
Develop a moisture detection system using ESP8266 and rain sensor. The system should send notification when moisture is detected.

17. ESP8266 IoT Water Quality Monitoring
Design a water quality monitoring system using NodeMCU and sensors for pH or turbidity measurement.

18. NodeMCU IoT Fire Detection System
Develop a fire detection system using smoke sensors and ESP8266. The system should send alerts over the internet.

19. ESP32 Health Monitoring System
Design a health monitoring system using ESP32 to measure pulse rate and body temperature and upload data online.

20. ESP32 Smart Dustbin
Create a smart dustbin using ESP32 and ultrasonic sensor where the lid opens automatically when a person approaches.

21. ESP32 Smart Water Level Monitoring
Design a wireless water tank monitoring system using ESP32 and ultrasonic sensor. Water level data should be accessible online.

22. ESP32 Smart Agriculture Monitoring
Develop an IoT farming system using ESP32 for monitoring soil moisture, humidity, and temperature.

Raspberry Pi-Based Design Problems

23. Raspberry Pi Smart Weather Station
Develop a weather station using Raspberry Pi and environmental sensors for monitoring atmospheric conditions.

24. Raspberry Pi IoT Smart Garden
Develop a smart gardening system using Raspberry Pi for monitoring soil moisture and controlling irrigation automatically.

25. Raspberry Pi Health Monitoring Dashboard
Design a health monitoring dashboard using Raspberry Pi to display patient sensor data in real time.

Suggested Text Books

- Real-Time Systems Design and Analysis by Phillip A. Laplante.
- Internet of Things: Architecture and Design Principles by Raj Kamal.
- Hands-On RTOS with Microcontrollers by Brian Amos.
- Programming Embedded Systems, O'Reilly by Michael Barr and Anthony Massa.
- The 8051 Microcontroller Based Embedded Systems by Manish K. Patel

DSCC-20-Th, Theory: Advanced DBMS
Credits - 03, Contact hours – 45, Marks-75

Advanced Relational Database Design and Normalization Review of relational database design; functional dependency; closure of functional dependency set; attribute closure; minimal cover; decomposition; lossless-join decomposition; dependency preservation; limitations of 3NF; Boyce-Codd Normal Form; multivalued dependency; Fourth Normal Form; join dependency; Fifth Normal Form; significance and limitations of higher normal forms.	**8 hrs**
Query Processing and Optimization Overview of query processing; parsing and translation of queries; evaluation of expressions; transformation of relational expressions; equivalence rules; cost estimation and cost calculation; access paths; selection operation; sorting; join algorithms; evaluation plan and selection of evaluation plan; basic idea of query optimization.	**7 hrs**
Transaction Management, Concurrency Control and Recovery Concept of transaction; ACID properties; transaction states; schedules; serializability; conflict and view serializability; recoverability; concurrent execution; need for concurrency control; lock-based protocols; two-phase locking; timestamp-based protocols; validation-based protocols; deadlock handling: prevention, detection and recovery; failure classification; log-based recovery; deferred and immediate database modification; checkpoints; shadow paging; buffer management and recovery.	**8 hrs**
Database System Architecture and Distributed Databases Overview of centralized database systems, client-server database systems, parallel database systems and distributed database systems; basic concepts of distributed databases; advantages and challenges of distributed databases; data fragmentation: horizontal, vertical and mixed fragmentation; replication and allocation; distributed database design; transparency in distributed databases.	**8 hrs**
Distributed Query and Transaction Management Distributed query management and access strategy: translation of global queries into fragment queries; query decomposition; data localization; join query processing; semi-join query processing; basic idea of distributed query optimization. Distributed transaction and concurrency management: introduction to distributed transactions; two-phase commit protocol; distributed concurrency control; timestamp-based distributed concurrency control; distributed deadlock detection and handling.	**8 hrs**
Data Warehousing and Emerging Database Systems Data warehousing: basic concepts of data warehousing; need and applications of data warehouse; data mart; OLTP vs. OLAP; dimensional modelling; star schema and snowflake schema; basic concept of ETL; introduction to data cube and OLAP operations. Emerging database systems: basic concepts and applications of spatial databases, statistical databases, temporal databases and real-time databases; introduction to NoSQL databases; key-	**6 hrs**

value database, document database, column-family database and graph database; introduction to CAP theorem; basic differences between SQL and NoSQL databases.	

DSCC-20-TU: Tutorial

Credits - 01, Contact hours – 30, Marks-25

Suggested Text Books

1. Database System Concepts, Abraham Silberschatz, Henry F. Korth and S. Sudarshan, McGraw-Hill.
2. Fundamentals of Database Systems, Elmasri and Navathe, Pearson.
3. Distributed Databases, Stefano CERI, Tata McGraw-Hill Education.
4. Database Systems: A Practical Approach to Design, Implementation and Management, Thomas Connolly and Carolyn Begg, Pearson.
5. An Introduction to Database Systems, C. J. Date, Pearson.