Sidho-Kanho-Birsha University, Purulia

Ranchi Road, PO-Sainik School Purulia

Dist- Purulia, West Bengal, India

PIN-723104

www.skbu.ac.in

Department of Computer Science

Syllabus for M.Sc. in Computer Science

with effect from the academic session 2020-21
**Semester I**

I. Principle of Programming Languages (Procedural, Functional, Logic, Object oriented)

II. Advanced Operating System (network, distributed, real-time, cloud)

III. Design and Analysis of Algorithms

IV. Mathematical Foundations (Statistical techniques, statistical inferences, linear algebra, tensor, optimization techniques)

V. Operating system Lab (Network OS, multi thread prog, Open source cloud)

VI. Programming & Algorithm Analysis Lab (java, python, functional programming)

**Semester II**

I. Formal Languages & Automata Theory

II. ADBMS (distributed, data warehouse, data----as per NET)

III. Advanced Computer Networks (fog, cloud, edge, sensor, wireless)

IV. Information Security (coding theory, cryptography, crypto analysis)

V. Network Lab

VI. Database Lab

**Semester III**

I. Compiler Design

II. Artificial Intelligence

III. Open Elective---- fixed

IV. Major Elective-I

V. AI Lab

VI. Compiler Design Lab
Semester IV

I. Data Analytics (statistical analysis, optimization, neural network, regression, mining, machine learning, )

II. Advanced Software Design (object oriented, uml, testing, verification, quality analysis)

III. Major Elective – II

IV. Major Elective – III

V. Add on Course--- Data Analytics/Advanced Software Design

VI. Major Project

VII. Seminar and Grand Viva

ELECTIVES:

Group-I: Data Science

Machine Learning, Deep Learning, Computer Vision & Pattern Recognition, Business Intelligence, Soft Computing

Group-II: Cyber Security

Network Security, Digital Forensic, Post Quantum Cryptography, Hardware Security, Cyber Law

Group-III: Distributed System & Resources

Cloud Computing, IOT, Service Oriented Computing, Semantic Web, Multimedia Systems & Database
<table>
<thead>
<tr>
<th>Paper Code</th>
<th>Paper Title</th>
<th>Credits</th>
<th>Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCSTCC101</td>
<td>Programming Languages</td>
<td>4</td>
<td>40+10</td>
</tr>
<tr>
<td>MCSTCC102</td>
<td>Advance Operating systems</td>
<td>4</td>
<td>40+10</td>
</tr>
<tr>
<td>MCSTCC103</td>
<td>Design and analysis of Algorithms</td>
<td>4</td>
<td>40+10</td>
</tr>
<tr>
<td>MCSTCC104</td>
<td>Mathematical foundations</td>
<td>4</td>
<td>40+10</td>
</tr>
<tr>
<td>MCSTCS105</td>
<td>Programming &amp; algorithm analysis Lab</td>
<td>4</td>
<td>50</td>
</tr>
<tr>
<td>MCSTCS106</td>
<td>Operating system Lab</td>
<td>4</td>
<td>50</td>
</tr>
</tbody>
</table>

(10 Marks: Internal Assessment, 40 Marks: Term Examination)

**Curriculum Semester I**

Paper Code – **MCSTCC101**

Paper Name – **Programming Languages**

<table>
<thead>
<tr>
<th>Module</th>
<th>Topics</th>
<th>Lecture Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language Design and Translation Issues</td>
<td>Programming Language Concepts, Paradigms and Models</td>
<td></td>
</tr>
<tr>
<td>Programming Environments</td>
<td>Binding and scope, Programming, Language Syntax and semantics, Stages in Translation, Formal Transition Models. Compilers, Interpreters, Interactive development tools, Debugging tools, variables, data abstraction,</td>
<td></td>
</tr>
<tr>
<td>Memory management</td>
<td>Static &amp; dynamic allocation, control structures, selective structures. Modular programming, function, parameter passing methods, lifetime of variables, recursion, error handling.</td>
<td></td>
</tr>
<tr>
<td>Elementary Data Types</td>
<td>Properties of Types and Objects; Scalar and Composite Data Types.</td>
<td></td>
</tr>
</tbody>
</table>
### Programming Languages

| Object oriented-programming, Event driven programming, Exception handling, Concurrent programming, Foundations of functional programming: \(\lambda\)-calculus, type checking, Logic programming, Scripting languages |
|---|---|

| Object Oriented Programming using Python |
|---|---|
| Characteristics of object oriented programming, Classes and Objects, Methods. Operator Overloading, super () and Method Overriding, abstraction, inheritance, polymorphism, Design Principles for object oriented programming, |

**Textbook:** 1. R. Sebesta, Concepts of Programming Languages, Addison Wesley

**References:**


5. Rambaugh, James Michael, Blaha – “Object Oriented Modelling and Design” – Prentice Hall, India


7. 

---

**Paper Code –** **MCSCCT102**

**Paper Name –** **Advance Operating systems**

<table>
<thead>
<tr>
<th>Module</th>
<th>Topics</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Credit:</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Full Marks:</th>
</tr>
</thead>
</table>

| 40+10 |

---

<table>
<thead>
<tr>
<th>Lecture Hours</th>
</tr>
</thead>
</table>

|  |  |
| Distributed file system | File service architecture - network file system- Andrew file system- recent advances Transactions and concurrency control: nested transactions-locks-optimistic concurrency control-comparison of methods for concurrency control-flat and nested distributed transactions-distributed deadlock-transactions recovery. Replication System model and group communication- fault tolerant services-transactions with replicated data | 16 |
| Real-Time | Real-time process concepts, categories of real-time task, real-time scheduling, | 10 |
| Networks | | 8 |
| Cloud Computing | Overview, Introduction to service and service oriented architecture, Layers and types of cloud service model, Uses of Cloud; Components of Cloud Computing - Software as a Service, Platform as a Service, Infrastructure as a Service, Identity as a Service; Data storage and virtualization in the cloud | 6 |

**Textbook:** 1. George Coulouris, Jean Dollimore and Tim Kindberg, Distributed Systems: Concepts and Design - Pearson Education

**References:** 2. Andrew S Tanenbaum and Marten Van Steen, Distributed Systems: Principles and paradigms – Pearson Education


<table>
<thead>
<tr>
<th>Module</th>
<th>Topics</th>
<th>Lecture Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>String Processing</td>
<td>KMP, Boyre-Moore, Rabin Karp algorithms.</td>
<td>4</td>
</tr>
<tr>
<td>Introduction to randomized algorithms</td>
<td>Random numbers, randomized quick sort, randomly built binary search tree.</td>
<td>3</td>
</tr>
<tr>
<td>Number Theoretic Algorithms</td>
<td>GCD, addition and multiplication of two large numbers, polynomial arithmetic, Fast-Fourier transforms.</td>
<td>3</td>
</tr>
<tr>
<td>Advanced Techniques to analyze algorithms</td>
<td>Use and study advanced data structures unionfind (Disjoint Set Structure), Fibonacci heaps.</td>
<td>3</td>
</tr>
<tr>
<td>Graph algorithms</td>
<td>Matching and Flows, Graph capture</td>
<td>5</td>
</tr>
<tr>
<td>Parallel algorithms</td>
<td>Basic techniques for sorting, searching and merging in parallel.</td>
<td>15</td>
</tr>
<tr>
<td>Geometric algorithms</td>
<td>Point location, Convex hulls and Voronoi diagrams.</td>
<td>8</td>
</tr>
<tr>
<td>Complexity Theory</td>
<td>P and NP Class Problems; NP-completeness and Reducibility.</td>
<td>4</td>
</tr>
</tbody>
</table>

**Textbook:** 1.T.H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein Introduction to Algorithms,
PHI, 3rd Edition 2009

**References:** 2. Sarabasse & A.V. Gelder, Computer Algorithm – Introduction to Design and Analysis, Publisher, Pearson 3rd Edition 199


<table>
<thead>
<tr>
<th>Paper Code – MCSCCT104</th>
<th>Paper Name – Mathematical Foundations</th>
<th>Credit: 4</th>
<th>Full Marks: 40+10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module</td>
<td>Topics</td>
<td>Lecture</td>
<td>Hours</td>
</tr>
<tr>
<td>Probability and Statistics</td>
<td>Discrete probability, simple random variables, the law of large numbers, Binomial, Poisson and normal distributions, central limit theorem, Non-uniform probability distributions, Bayes’ Theorem on conditional probability. Importance of random sampling, sampling distributions (Normal, t-distribution, chi-square distribution, F-distribution, ANOVA); statistical inferencing. Rejection method, Metropolis algorithm, random variables, expectations and moments, Stochastic process, Markov model</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Sets and Relations</td>
<td>Permutation Functions, Growth of Functions. Partially ordered sets, Lattices, Finite Boolean algebra.</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Algebraic structures</td>
<td>Semigroups, Monoids, Groups, Subgroups, Symmetric groups, Groups homomorphism and isomorphism, Cosets</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>
and Lagrange’s Theorem, Normal subgroups, Permutation of groups and Burnside’s theorem. Boolean Functions and its Representation, Simplifications of Boolean Functions.

<table>
<thead>
<tr>
<th>Graph Theory</th>
<th>Basic Concept of Graph Theory, Euler Paths and Circuits, Hamiltonian Paths and Circuits, Spanning tree.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimization</td>
<td>Linear Programming Problems - Mathematical Model, Graphical Solution, Simplex and Dual Simplex Method, Sensitivity Analysis.</td>
</tr>
<tr>
<td></td>
<td>Integer Programming, Transportation and Assignment Models.</td>
</tr>
<tr>
<td></td>
<td>PERT-CPM: Diagram Representation, Critical Path Calculations, Resource Levelling, Cost Consideration in Project Scheduling.</td>
</tr>
<tr>
<td></td>
<td>Introduction to nonlinear programming problems</td>
</tr>
</tbody>
</table>

**References:**


2. Bernard Kolman c, Busby & Sharon Ross, Discrete Mathematical Structures


## Module | Topics | Lab Hours
---|---|---
**Python Programming** | Running Python Programs and User Interaction, Variables and Expressions, Data Types in Python, Advanced Data Types (List, Set, Tuples, related operations, Dictionary) |  
**Conditional expressions and Loops** | Conditional Expressions, Loops, Nested Loops, Exception Handling |  
**Functions, Recursions** | Syntax and Basics of a Function, Use of a function, Parameters and Arguments, Return statement, Local and Global Scope Scope of a Variable, Recursive functions |  
**File Handling** | Need of File Handling, Reading/Writing Text and Numbers to/from a File, Directories on a disk. |  
**Modules and packages** | Understanding and Creating Python modules and packages for modular programming, NumPy, Pandas, MatPlotLib, and NLTK |  
**Object Oriented Programming** | Classes and Objects, Methods. Operator Overloading, Inheritance, super 0 and Method Overloading. |  
**Programming using Python** | • Function definitions: pattern matching, induction  
• Basic data types, tuples, lists  
• Higher order functions  
• Polymorphism  
• Reduction as computation, lazy evaluation  
• Measuring computational complexity  
• Basic algorithms: sorting, backtracking, dynamic programming  
• User-defined datatypes: enumerated, recursive and |
Design and Analysis of Algorithms Lab

<table>
<thead>
<tr>
<th>polymorphic types</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Input/output</td>
</tr>
</tbody>
</table>

1) i. Implement Insertion Sort (The program should report the number of comparisons)

ii. Implement Merge Sort (The program should report the number of comparisons)

2. Implement Heap Sort (The program should report the number of comparisons)

3. Implement Randomized Quick sort (The program should report the number of comparisons)

4. Implement Radix Sort

5. Create a Red-Black Tree and perform following operations on it: i. Insert a node ii. Delete a node iii. Search for a number & also report the color of the node containing this number.

6. Write a program to determine the LCS of two given sequences

7. Implement Breadth-First Search in a graph

8. Implement Depth-First Search in a graph

9. Write a program to determine the minimum spanning tree of a graph


References: 2.

3.
### Module | Topics | Lab Hours
--- | --- | ---
Distributed System |  |  
Open source cloud |  |  
Multi-threaded programming |  |  
Network operating system |  |  

**Textbook:** 1.

**References:** 2.
3.
Semester 2

Paper 201: Formal Languages and Automata Theory

Mathematical Preliminaries for Finite Automata and Formal Languages:
Sets, Relations and Functions, Graphs and Trees, Mathematical Induction, Mathematical Logic, Formal Language, Chomsky Hierarchy Languages (CHL), Automata Theory

Regular Expressions and Regular Languages:
Regular Expressions, Regular Languages and Regular Grammar

Finite State Automata:
Deterministic Finite Automata, Non-Deterministic Finite Automata, NFA and Regular Expressions, Conversion of Finite Automata to Regular Expression, Conversion of NFA to DFA, NFA with e-Transitions (e-NFA), Conversion from e-NFA to NFA, Conversion from e-NFA to DFA, Output Associated with Finite Automata, Moore and Mealy Machines, Minimization of Automata

Properties of Regular Languages:
Regular sets, regular expressions, identity rules, Constructing finite Automata for a given regular expressions, Conversion of Finite Automata to Regular expressions. Pumping lemma of regular sets, closure properties of regular sets (proofs not required).

Grammar Formalism: Regular grammars-right linear and left linear grammars, equivalence between regular linear grammar and FA, inter conversion, Context free grammar, derivation trees, sentential forms. Right most and leftmost derivation of strings.


Push Down Automata: Push down automata, definition, model, acceptance of CFL, Acceptance by final state and acceptance by empty state and its equivalence. Equivalence of CFL and PDA, interconversion. (Proofs not required). Introduction to DCFL and DPDA.

Turing Machines:
Turing Machine, definition, model, design of TM, Computable functions, recursively enumerable languages. Church's hypothesis, counter machine, types of Turing machines (proofs not required).
**Computability Theory:** Chomsky hierarchy of languages, linear bounded automata and context sensitive language, LR(0) grammar, decidability of, problems, Universal Turing Machine, undecidability of posts. Correspondence problem, Turing reducibility, Definition of P and NP problems, NP complete and NP hard problems.

**References**
- “Introduction to Automata Theory Languages and Computation”. Hopcroft H.E. and Ullman J. D.Pearson Education
- Introduction to Theory of Computation –Sipser 2nd edition Thomson
- Introduction to languages and the Theory of Computation ,John C Martin, TMH
- Theory of Computer Science – Automata languages and computation -Mishra andChandrashekaran, 2nd edition, PHI

**Paper 202: Advanced DBMS (distributed dbms, recovery management, data warehouse, data mining, no-SQL, follow NET syllabus)**

**Database System Concepts and Architecture:** Data Models, Schemas, and Instances; Three-Schema Architecture and Data Independence; Database Languages and Interfaces; Centralized and Client/Server Architectures for DBMS. (4)

**Data Modeling:** Relational Schemas, Update Operations and Dealing with Constraint Violations; Relational Algebra and Relational Calculus; Codd Rules. (8)

**SQL:** Views, Stored Procedures and Functions; Database Triggers, SQL Injection. (8)

**Normalization and Transactional Management:** Functional Dependencies and Normalization; Algorithms for Query Processing and Optimization; Transaction Processing, Concurrency Control Techniques, Database Recovery Techniques, Object and Object-Relational Databases; Concepts of ACID and CAP theory, Database Security and Authorization. Transactions, Recovery, System Recovery, Media Recovery, Two-phase Commit. (14)

**Enhanced Data Models:** Temporal Database Concepts, No SQL databases- document oriented models, key-value models, columnar data models, graph data model, Multimedia Databases, Deductive Databases, link data management. (10)

**Distributed Database:** Distributed Database System, Distributed Database Design, Data Fragmentation, Data Replication, Data Allocation, Query Processing in Distributed Databases. (8)
Data Warehousing: Data Modeling for Data Warehouses, Concept Hierarchy, OLAP and OLTP. (4)

Big Data Systems: Big Data Characteristics, Types of Big Data, Big Data Architecture, Introduction to Map-Reduce and Hadoop; Distributed File System, HDFS. (4)

References

- Silberschatz, Korth, Database System Concepts, TMH
- C.J. Date, Database Design and Relational Theory: Normal Forms and All That Jazz

Paper 3: Advanced Computer Networks
(Distributed networks, IOT, fog, cloud, wireless sensor networks, internet working, ssl)

Data Communication: Components of a Data Communication System, Digital and Analog Transmission; Data Encoding and Modulation Techniques; Broadband and Baseband Transmission; Multiplexing, Transmission Media, Transmission Errors, Error Handling Mechanisms. (8)


IPv4 Structure and Address Space; Classful and Classless Addressing; Datagram, Fragmentation and Checksum; IPv6 Packet Format, Mapping Logical to Physical Address (ARP), Direct and Indirect Network Layer Delivery; Routing Algorithms, TCP, UDP and SCTP Protocols; Flow Control, Error Control and Congestion Control in TCP and SCTP. (20)

**Internetworking:** Switch/Hub, Bridge, Router, Gateways, Concatenated virtual circuits, Tunnelling, Fragmentation. (8)

**Network Security:** Malwares, Cryptography and Steganography; Secret-Key Algorithms, Public-Key Algorithms, Digital Signature, Virtual Private Networks, Firewalls, Mechanisms of TLS, SSL, IPsec. (16)

**Adhoc Networks:** Mobile Adhoc Networks, P2P network, Wireless Transmission and Wireless LANs, semantic sensor networks. (8)

**Cloud Computing and IoT:** SaaS, PaaS, IaaS, Public and Private Cloud; Virtualization, Virtual Server, Cloud Storage, Database Storage, Resource Management, Service Level Agreement, Basics of IoT.

**References**
- Data Communications and Networking — Behrouz A. Forouzan, Third Edition TMH.
- Understanding communications and Networks, 3rd Edition, W.A. Shay, Thomson COMPUTE

**Paper 4: Information and Coding Theory**

**Models for Information Channel:** Discrete Memoryless Channel, Binary Symmetric Channel (BSC), Burst Channel. Bit-error rates. Probability, Entropy and Shannon’s measure of information. Mutual information. Channel capacity theorem. Rate and optimality of Information transmission. (10)

**Variable Length Codes:** Prefix Codes, Huffman Codes, Lempel-Ziev (LZ) Codes. Optimality of these codes. Information content of these codes, Vcoder. (8)

**Error Correcting and Detecting Codes:** Finite fields, Hamming distance, Bounds of codes, Linear (Parity Check) codes, Parity check matrix, Generator matrix, Decoding of linear codes, Hamming codes, decoder, BCH. (12)

**Cryptography:** Concepts and Techniques, symmetric and asymmetric key cryptography, steganography, Symmetric key Ciphers: DES structure, DES Analysis,
Security of DES, variants of DES, Block cipher modes of operation, AES structure, Analysis of AES, Key distribution Asymmetric key Ciphers: Principles of public key cryptosystems, RSA algorithm, Analysis of RSA, Diffie-Hellman Key exchange. (10)

**Message Authentication and Hash Functions:** Authentication requirements and functions, MAC and Hash Functions, MAC Algorithms, MD5: Secure Hash Algorithm, Whirlpool, HMAC, Digital signatures, X.509, Kerberos, IPSec, Secure Socket Layer(SSL), Transport Layer Security(TLS) (10)

**Introduction to Cryptoanalysis:** Linear Cryptanalysis, Differential Cryptanalysis, Cryptanalysis of DLP (5)

Recent trends in security: IOT, Biometric, Visual Cryptography. (5)

Text Books:

- Cryptography and Network Security; McGraw Hill; Behrouz A Forouzan
- Information Security Intelligence Cryptographic Principles andApp. Calabrese Thomson
- D. P. Nagpal, Information Security, S. Chand Company Limited

**Paper 5: Network Lab**

Problems and assignments related to Paper 3.

**Paper 6: Database Lab**

Problems and assignments related to Paper 2.