Syllabus for B.Sc Computer Science(Hons.)
[2016 onwards]

Sikkim University
Gangtok, Sikkim

Syllabus Drafting Committee

1. Dr. Subir Mukhopadhyay, Associate Professor, Department of Physics, Sikkim University
Chairperson

2. Dr. Asoke Nath, Associate Professor, Department of Computer Science, St. Xavier's College(Autonomous), Kolkata
Member

3. Mrs. Chunnu Khawas, Asst. Prof., Department of Computer Applications, Sikkim University
Member

4. Mrs. Rebika Rai, Asst. Professor, Department of Computer Applications, Sikkim University
Member

5. Mr. Partha Pratim Ray, Asst. Prof., Department of Computer Applications, Sikkim University
Member

6. Dr. Rakesh Kr. Ranjan, Assistant Professor, Department of Geology, Sikkim University
Member

Syllabus for B.Sc. Computer Science (Hons) (3 years)

Semester I			
Code	Paper	Credit	Details
CSC-UG-O101	Fundamental of Computer and C Programming	4	3T+1P
	Elective I	4	
	Elective II	4	

Semester II			
Code	Paper	Credit	Details
CSC-UG-O201	Data Structure using C	4	3T+1P
	Elective III	4	
	Elective IV	4	

Semester III			
Code	Paper	Credit	Details
CSC-UG-O301	Operating System and Digital Design	4	4T
	Elective V	4	
	Elective VI	4	

Semester IV			
Code	Paper	Credit	Details
CSC-UG-C401	Programming using C++ and Java	4	2T+2P
CSC-UG-C402	Discrete Mathematics	4	4T
	Compulsory I	4	

Semester V			
Code	Paper	Credit	Details
CSC-UG-C501	Formal Language and Automata Theory	4	4T
CSC-UG-C502	Database Management System	4	3T+1P
	Compulsory II	4	

Semester VI			
Code	Paper	Credit	Details
CSC-UG-C601	Software Engineering	4	3T+1P
CSC-UG-C602	Computer Networks and Project	4	3T+1P
	Compulsory III	4	

Abbreviations:

O - Open Paper (Compulsory for B.Sc. Computer Science students and Elective for any other Departments)

C - Compulsory Paper (Compulsory for B.Sc. Computer Science students)

T - Theory

P - Practical

Notes:

a. Compulsory I, II, and III may be opted from Environmental Studies, Eastern Himalayan Studies, and English.

b. One Practical class should be of 2 hours. Laboratory file should be maintained by the student.

c. Project work in the Paper CSC-UG-C602 is designed to enhance the real life problem solving skills among the students. Project is 1 credit course. As per convention, the project work done at the end of course always improves the visibility and credibility, especially for Computer background students.

   The project should be done under the supervision of the teacher. The project may be carried out on the subjects taught during the course. The marks allotted for Project shall be 25 marks (Project report submission: 20 and Viva: 5). Project viva should be conducted in presence of external subject expert.

CSC-UG-O101 Fundamentals of Computer and C Programming

UNIT I

Introduction: Characteristics and capabilities of a computer, Generation of Computers, Types of Computers, Memory and its types, Various input and output devices, Storage Devices: Storage Fundamentals: Primary Vs Secondary, Data Storage and Retrieval Method: Sequential, Direct and Index Sequential, Various storage devices, Types of software, Programming Languages.

Introduction to Digital Logic: Basics of Gates (AND, OR, NOT, XOR, NAND, NOR etc.).

Binary Number System & Boolean Algebra, BCD, ASCII, EBDIC, Gray codes and their conversions; Signed binary number representation with 1's and 2's complement methods, Venn diagram.

UNIT II

Overview in C: History of C, Importance of C, Basic structure of a C program, Different sample programs, Programming style, Executing a C program, Constants, variables, data types, Storage classes, Overflow and under flow of data.

Operators and expressions: Introduction, Different categories of operators in C language, Type conversions in expression, Operator precedence and associativity.

Looping and branching: Decision making and branching, Decision making and looping, Jumps in loops.

UNIT III

Arrays: 1-dimensional arrays, 2-dimensional arrays declarations and use; Introduction, Declaration & Initialization of an array, Types of arrays, Dynamic arrays, sorting techniques : Bubble Sort, Insertion Sort, Selection Sort, Merge Sort methods, Searching algorithms : Linear search method, Binary search method.

String processing: To find length, to concatenate stings, to extract words from a string, pattern matching in a string

User-defined functions and Pointers: Introduction, function definition, need for a user defined functions, elements of a user defined function, category of functions, nesting of functions, recursion, call by value and call by address, passing arrays to function, passing strings to a function, Pointers and handilng.

Preprocessor and Storage class: Preprocessor directive and its handling techniques, Different storage classes.

Structure, Union: Introduction, defining a structure, declaring structure variables, accessing structure members, array of structures, arrays within structures, unions.

File handling: different types of files such as Sequential, random access, Indexed sequential mode, Different functions to process a file such as fopen, FILE statement, fscanf(), fgetc(), fread(), fprintf(), fputc(), fwrite(), fseek(); Copying files, to split a file, extract words from a file, To change upper case to lower case and vice – versa, To search a pattern in a file and to replace it by a new pattern.

UNIT IV

Laboratory Experiments (See Annexure I)

References

1. WilliamS. Davis, Fundamental Computer Concepts, AddisonWesley, 1989.
2. E. Balaguruswamy, Programming in ANSI C, Tata McGraw-Hill, 2012.
3. K. Venugopal,Mastering in C, 1997.
4. Gottfried, Fundamentals of C, 2E, Schaum's Outlines 1996.
5. Kerninghamand Ritchie, the C Programming, 1990.
6. Y. Kanetkar, Let Us C, 12E, BPB, 2006.

CSC-UG-O201 Data Structure using C

UNIT I

Introduction to Data Structures: Definition, Classification of data structures (Linear and Non-Linear), Operations on data structures. Time Space Tradeoff, Asymptotic Notations, Conditional asymptotic notation, Removing condition from the conditional asymptotic notation-Properties of big-Oh, big-Omega, big-Theta, small-Oh, small-Omega notation, Recurrence equations, Solving recurrence equations.

String Processing: Strings, Storing Strings, Fixed length structures, Variable length structures with fixed maximums and linked structures-Primitive Operations on Strings-Substring, Indexing, Concatenation and Length of the string, Pattern Matching Algorithms: First Pattern Matching Algorithms and Fast Pattern Matching Algorithms.

Arrays: Definition, Representation of Linear arrays in memory (Both Single and Two Dimensional arrays), Algorithm for Insertion and Deletion in one dimensional arrays (ordered and unordered arrays), advantages and disadvantages of arrays, Sparse Matrices, Linear Search and Binary Search.

UNIT II

Linked Lists: Representation of linked lists in memory-Operations on linked list: Add Node, search node, display node, modify node, sort linked list, insert node before or after a node, reverse a linked list .

Circular linked lists (Insertion, Deletion, Display), doubly linked linear list (Insertion, Deletion, Display): Applications of linked linear lists: Addtion of two large prime numbers, To add, subtract and to take product of two polynomials.

Stacks: Concepts, Operations, sequential and linked implementation, Application of stacks: Towers of Hanoi, Infix, Prefix and Postfix expressions and Evaluation of postfix expression using stacks.

UNIT III

Queues Concepts, operations, sequential and linked implementation, Linear Queue (FIFO), Circular queues, Priority Queue, and application of queues.

Trees: Binary trees, Complete Binary tree, Binary Search Trees: Searching, Inserting and deleting in Binary Search Trees, Traversals on a BST (In-order, post-order, pre-order, DFS, BFS), Application of Trees.

Different Sorting methods and calculation of Time complexity: Merge sort, Quick sort, Heap Sort, Radix Sort.

Hashing: Definition, Advantages, Different Hash Functions, Collision Resolution Techniques, Applications

UNIT IV

Laboratory Experiments (See Annexure II)

References

1. T. H. Cormen, C. L. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, MIT Press, 2001.
2. J. Kleinberg and E. Tardos, Algorithm Design, Addison-Wesley, 2006.
3. Harry R. Lewis and Larry Denenberg, Data Structures and Their Algorithms, Harper Collins. 1991.
4. A. Gibbons, Algorithmic Graph Theory, Cambridge University Press, 1985.
5. Michael T. Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples, JohnWiley, 2006.
6. R. Sedgewick, Algorithms in C: Part 5, AddisonWesley, 2001.
7. M. H. Alsuwaiyel, Algorithm Design Techniques and Analysis,World Scientific, 1999.
8. Gilles Brassard and Paul Bratley, Algorithmics: theory and practice, Prentice-Hall, 1996.
9. Udi Manber, Introduction to Algorithms: A Creative Approach, Addison-Wesley, 1989.
10. Horowitz and Sahni – Fundamentals of Data Structures in C – Orient Longman Pvt. Ltd.
11. Reema Thareja – Data Structures using C – Oxford Publications
12. Srivastava and Srivastava – Data Structures Through C in Deprth – BPB Publications
13. Data Structure in "C", Horowitz & Sahni, Silicon Press
14. Data Structures & Program Design in "C", R. Kruse, Pearson Education
15. Data Structures using "C", A. M. Tenenbaum, Pearson Education
16. Data Structures with "C", Lipschutz, TMH

CSC-UG-O301 Operating System and Digital Design

UNIT I

Introduction: Basic concepts, Simple Batch Systems, Multi-programmed Batched Systems, Time-Sharing Systems, Protection.

Processes and CPU scheduling: Process Concept, Process scheduling, Operation on Processes, Cooperating Processes, Inter process Communication.

Scheduling: Scheduling criteria, Scheduling algorithms-FIFO, LRU, RR, and advanced algorithms.

Virtual Memory: Demand paging, Page replacement, Page-replacement algorithms.

Process Synchronization: The Critical-Section problem, Synchronization hardware, Basics of Semaphores.

UNIT II

Deadlocks: Deadlock characterization, Methods of Handling Deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection & Recovery from Deadlock.

Memory Management: Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging.
Multiprocessor Operating System: The thread concept, thread system calls, Uses of threads, Lightweight processes and user threads, examples of threads.
Preliminaries on Real Time OS and Embedded OS.

Digital Design
UNIT III
Boolean algebra and Gate Networks: Fundamental concepts of Boolean algebra, different Gates, universal Gates, Basic laws of Boolean algebra, DeMorgan's theorems, Simplification of Boolean expression, Karnaugh map (SOP and POS) with examples.
Combinational circuits - Adder and Subtractor circuits (half & full adder & subtractor); Encoder, Decoder, Comparator, Multiplexer, De-Multiplexer and Parity Generator.

UNIT IV
Sequential Circuits - Basic Flip-flop & Latch, Flip-flops -SR, JK, D, T and JK Master-slave Flip Flops Registers (SISO,SIPO,PIPO,PISO), Ring counter, Johnson counter, Basic concept of Synchronous and Asynchronous counters (detail design of circuits excluded), Design of Mod N Counter

References
1. Silberschatz and Galvin, Operating System Concepts, AdditionWesley, 1999.
2. H.M.Dietel, An Introduction to Operating System, Pearson Education,.
3. Modern Operating Systems; Andrew S. Tanenbaum, Prentice Hall of India
4. William Stallings, "Operating Systems: Internals and Design Principles ", 6th Edition, Pearson Education
5. Operating Systems: Principles and Design; Pabitra Pal Choudhury, PHI Learning
6. David A. Patterson and John L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, Elsevier, 2012.
7. Carl Hamachar, Zvonco Vranesic and Safwat Zaky, Computer Organization,McGraw Hill, 2002.
8. John P. Hayes, Computer Architecture and Organization,McGrawHill, 1998.
9. WilliamStallings, Computer Organization and Architecture: Designing for Performance, Pearson Education., 2007.
10. Vincent P. Heuring and Harry F. Jordan, Computer Systems Design and Architecture, Pearson Education, 2008.
11. Digital Computer Electronics, Malvino and Brown, Tata McGraw-Hill
12. Digital Logic and Computer Design, M Morrs Mano, Pearson education India

CSC-UG-C401 Programming using C++ and Java

UNIT I
Concepts of OOP: Introduction OOP, Procedural Vs. Object Oriented Programming, Principles of OOP, Benefits and applications of OOP.
C++ Basics:

Overview, Program structure, namespace, identifiers, variables, constants, enum, operators, typecasting, control structures.

Basic Concepts on C++ Functions: Simple functions, Call and Return by reference, Inline functions, Macro, Overloading of functions, friend functions, virtual functions.

Basic Concepts on Objects and classes: Basics of object and class in C++, different members, constructors, destructors, operator overloading.

Basic Concepts on Inheritance: Types of inheritance, overriding, virtual base class.

Basic Concepts on Polymorphism: Pointers in C++.

Java Basics:

History of Java, Java buzzwords, data types, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java program, concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, parameter passing, recursion, nested and inner classes, exploring string class.

UNIT II

Inheritance: Hierarchical abstractions, Base class object, subclass, subtype, substitutability, forms of inheritance- specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance. Member access rules, super uses, using final with inheritance, polymorphism- method overriding, abstract classes, the Object class.

Exception handling: Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception sub classes. String handling,Exploring java.util.

Basic of Applets.

Multi Threaded programs in Java, Sleep(), Stop(), Yield() , run(), start() functions, Thread Class, priority of Threads, applications of Multi Threaded programs in Java.

File Handling in Java, Sequential file, random access file.

UNIT III

Laboratory Experiments Related to C++ (See Annexure III)

UNIT IV

Laboratory Experiments Related to Java Laboratory (See Annexure IV)

References

1. Schildt, Herbert, C++ (Computer program language), McGraw-Hill, New York: 2008.

2. Savitch, Walter Problem solving with C++: The object of programming/ C++, Pearson Addison Wesley, 2005.

3. Malik, D S, C++ programming: From Problem Analysis to Program Design / C plus plus programming, Course Technology, 2009.

4. H. Schildt, Java: The Complete Reference, TMH, 2005.

5. T. Budd, Understanding OOP with Java, Pearson Eduction, 2007.
6. J. Nino, F. A. Hosch, An Introduction to programming and OO design using Java, John Wiley & Sons, 2004.
7. T. Budd, An Introduction to OOP, Pearson Education, 2006.
8. Y. D. Liang, Introduction to Java programming, Pearson Education, 2007.
9. Programming using Java : E. Balagurusamy, Tata McGraw Hill Company.

CSC-UG-C402 Discrete Mathematics

UNIT I

Logic and Proof, Sets and Functions: Propositional Equivalences, Predicates and Quantifiers, Methods of Proof.

Set Operations :
Superset, subsets, union, Intersection, difference, symmetric difference, Venn diagrams, Cartesian product. Relations as a subset of a product. Binary relations. Functions , growth of function Reflexive, symmetric and transitive properties of a relation on a set. Closure relations, Equivalence relations, partial orderings. Examples.

UNIT II

Counting Theory: Basics of Counting, Permutations and Combinations, Pigeon Hole Principle, Recurrence Relation, Generating Function, Inclusion and Exclusion Principle. Principle of mathematical induction.

UNIT III

Introduction to Probability Theory: Introduction to discrete probability, sample space, Finite probability space, Conditional probability, Independence, Independent repeated trials, Bernoulli Trials and Binomial distribution, probability distribution of Random variable, Expectation of random variable, Variance.

UNIT IV

Graph Theory: Definition of Graph, Graph Terminology, Finite and Infinite graphs. Directed and undirected graphs, Degree, Isolated vertex, Pendant vertex. Null graphs. Walks: Paths and circuits. Connected and disconnected graphs, Euler's graphs, Hamiltonian paths and circuits. Planer Graph, Isomorphic graph, coloring graphs Graph algorithms : Adjacency Matrix, Warshall algorithm, Floyd’s shortest Path algorithm, Trees, Shortest spanning tree using Kruskal Algorithm and Prim’s Algorithm, Dijkstra’s Algorithm to find shortest path from a given vertex. Graph Search: Breadth First Search Algorithm(BFS), Depth First Search Algorithm Problems(DFS).

References

1. Discrete Mathematics and its Applications : Kenneth H.Rosen : Publ: Tata McGraw-Hill Publishing Company Limited.
2. Elements of Discrete Mathematics : C.L.Liu: Publ: Tata McGraw-Hill Publishing Company Limited.
3. Discrete Mathematical Structures : Kolman, Busby, Ross : Publ: Pearson Education.
4. Enginering Mathematics : B.K.Pal, K.Das : U.N.Dhar & Sons Pvt. Ltd, Kolkata.
5. Graph Theory : Narshing Deo .
6. Introduction to Graph Theory : Douglas B.West : Publ: PHI:

CSC-UG-C501 Formal Language and Automata Theory

UNIT I

Introduction: Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages.

Regular languages and finite automata: Regular expressions and languages, deterministic finite automata (DFA) and equivalence with regular expressions, nondeterministic finite automata (NFA) and equivalence with DFA, regular grammars and equivalence with finite automata, properties of regular languages, pumping lemma for regular languages, minimization of finite automata.

Finite State Automata with Output, Moore and Mealy machines.

UNIT II

Context-free Languages and Pushdown Automata: Context-free grammars (CFG) and languages (CFL), Chomsky and Greibach normal forms, nondeterministic pushdown automata (PDA) and equivalence with CFG, parse trees, ambiguity in CFG, pumping lemma for context-free languages, deterministic pushdown automata, closure properties of CFLs.

Context-sensitive languages: Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG.

UNIT III

Turing Machines: The basic model for Turing machines (TM), Turing-recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, variants of Turing machines, nondeterministic TMs and equivalence with deterministic TMs, unrestricted grammars and equivalence with Turing machines, TMs as enumerators.

UNIT IV

Undecidability: Church-Turing thesis, Universal Turing machine, the universal and diagonalization languages, reduction between languages and Rice's theorem, undecidable problems about languages.

References

1. Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation, Pearson Education Asia, 1988.
2. John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, Pearson Education Asia, 2008.
3. Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer, 1997.
4. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing, 2012.
5. John Martin, Introduction to Languages and The Theory of Computation, Tata McGraw Hill, 2010.
6. "Introduction to Formal languages and Automata", Peter Linz, Narosa Publishing House
7. Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.
8. Theory of Computer Science – Automata languages and computation -Mishra and Chandrashekaran, 2nd edition, PHI.

CSC-UG-C502 Database Management System

UNIT I

Introduction: Database management system, Characteristics of the database approach, Actors on the scene, Workers behind the scene, Advantages of Using a DBMS and when not to use a DBMS.

DBMS Architecture: Data Models: Categories of data models, Database state, DBMS Architecture and Data Independence, DBMS architecture, DBMS Languages and Interfaces, Classifications of Database Management Systems.

Data Modeling Using E-R Model: Using High Level Conceptual Data Models for Database Design, Example Database applications, Entity Sets and types, Attributes and Keys, Relationship and its types , Constraints, Designing E- R Diagrams, Mapping E-R diagram to relations.

UNIT II

Index Structures for Files: Single Level Ordered Indexes, Primary indexes, Clustering indexes and Secondary indexes. Multi-level indexes, Dynamic Multilevel indexes using B-trees (Introductory concepts), hashing concepts.

Relational Data Model: Relation, Integrity constraints, Basic Relational algebra operations, Functional dependencies, Normalization for Relational Databases: Normalization concepts, first, second, third and Boyce-Codd normal form.

UNIT III

SQL: (DDL/ DML), Queries, sub queries, updation of a database through views, Update, Delete.

Transaction Processing Concepts and Concurrency Control Techniques: Transaction and System Concepts, Desirable properties of Transactions (ACID), Schedules and Recoverability, Lock-Based Protocols: Locks, Granting of Locks, and Two phase locking protocol and implementation of locking.

Distributed Databases: Introduction; Comparison with traditional databases; Reference Architecture; DDBMS Components; Access Methods; Fragmentation, Replication,Allocation.

UNIT IV

Laboratory Experiments (See Annexure V)

References

1. Abraham Silberschatz, Henry Korth, and S. Sudarshan, Database System Concepts, McGraw-Hill, 2010.
2. Raghu Ramakrishnan, DatabaseManagement Systems,WCB/McGraw-Hill, 2003,
3. J. D. Ullman, Principles of Database Systems, Galgotia, 1985.
4. R. Elmasri and S. Navathe, Fundamentals of Database Systems, Addison-Wesley, 2008.
5. Bipin Desai, An Introduction to Database Systems, Galgotia, 1990.
6. Serge Abiteboul, Richard Hull and Victor Vianu, Foundations of Databases. Addison-Wesley, 1995.
7. An introduction to Database Systems by, C.J.Date, Narosa Publications

CSC-UG-C601 Software Engineering

UNIT I

Introduction to Software Engineering: Characteristics, Emergence of Software Engineering, Software Metrics & Models, Process & Product Metrics.

Software Life Cycle Models: Discussion on SDLC, Waterfall, Prototype, RAID and Spiral Models and their Comparison.

UNIT II

Software Project Management: Size Estimation- LOC and FP Metrics, Cost Estimation- Delphi and Basic COCOMO, Staffing Level Estimation, Putnam's Model.

Software Requirements Specification: SRS Documents, their Characteristics and Organization.

UNIT III

Software Design: Classification, Software Design Approaches, Function Oriented Software Design, Structured Analysis - Data flow Diagrams and Structured Design, Introduction to Object Oriented Design.

Coding and Testing of Software: Unit Testing, Block Box Testing, White Box Testing, Debugging, Program Analysis Tools, System Testing, Coding Standards and Guidelines.

UNIT IV

Laboratory Experiments (See Annexure VI)

References

1. Rajib Mall, Fundamentals of Software Engineering, Prentice Hall of India, 2005.
2. Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Pub, 2005.
3. Richard Fairley, Software Engineering Concepts, Tata McGraw Hill, 2006.
4. Roger S. Pressman, Software Engineering: A practitioner's approach, McGraw Hill, 2005.
5. Software Engineering, Ian Sommerville - Pearson Education
6. Object-Oriented Analysis and Design with Applications, Grady Booch, Robert A. Maksimchuk, Addison Wesley

CSC-UG-C602 Computer Networks and Project

UNIT I

Introduction: Introduction of network, topology, Use of computer network, network hardware: LAN, WAN, MAN, Wireless Network, Reference Models: ISO-OSI model, TCP model.

Physical layer: Transmission media-Magnetic Media, Twisted Pair, Coaxial pair, Fiber Optics, Line coding, Multiplexing, Spread spectrum.

Mac sub layer: Multiple Access protocol: ALOHA, Slotted ALOHA, CSMA protocols, Introduction to MAC Protocols: 802.3, 802.4, 802.5, 802.11

UNIT II

Data link layer: Data link layer design Issue, Error Detection and correction, Elementary Data link protocol, stop and-wait ARQ, sliding window, Go-back-n, Selective Repeat ARQ. Related advanced algorithms to be studied.
Network layer: Network Design Issue, Routing algorithm-introduction, optimality Principle, Shortest Path, Flooding, Distance Vector Routing. Congestion Control Routing: General principle of Congestion control, leaky bucket algorithm, Token Bucket Algorithm.

UNIT III

TCP/IP: The TCP/IP architecture, the Internet Protocol, ARP, DHCP and mobile IP, Internet routing protocols-RIP, OSPF, BGP. TCP/IP Implementation related case studies to be studied.
Transport layer: Transport Services, Element of transport protocols, TCP connection management, TCP transmission policy, TCP congestion control & Timer management.
Application layer: DNS, SMTP, POP3, FTP, TELNET, HTTPS. Related advanced protocols should be studied.

UNIT IV

Project

Project will be assigned in 5th semester and to be implemented within 6th semester. The project synopsis should be approved within 5th semester. Project should be supervised by internal/external faculty members. Project work may be carried out in the parent institution or research/academic institutes. Project will be evaluated in the 6th semester.

References

1. William Stallings, Data and Computer Communication, Prentice Hall of India, 2007.
2. Behrouz A. Forouzan, Data Communication and Networking,McGraw-Hill, 2007.
3. Andrew S. Tanenbaum, Computer Networks, Prentice Hall, 2008.
4. Douglas Comer, Internetworking with TCP/IP, Volume 1, Prentice Hall of India, 2006.
5. W. Richard Stevens, TCP/IP Illustrated: The Protocol, Volume 1, Addison-Wesley, 2011.
6. William Stallings, Cryptography and Network Security: Principles and Practice, Prentice Hall of India, 2008.
7. Neal Koblitz, A course in number theory and cryptography, Springer, 2008.
8. R. C. Seacord, Secure Coding in C and C++, Addison-Wesley, 2005.
9. John Viega, Matt Messier), Pravir Chandra, Network Security with OpenSSL, O’Reilly, 2009.
10. John Viega, Matt Messier, Secure Programming Cookbook for C and C++: Recipes for Cryptography, Authentication, Input Validation & More, O’Reilly, 2009.

Annexure I

CSC-UG-O101

Following problems in C programming should be done using Linux environment and GCC compiler. At least 5 problems should be done from each group in the following.

Group A

1. Input any positive integer ‘N’ which should be less than or equal to 20. Calculate and print the following :

   i. Sum = 1+2+3+…………+N

   ii. Sum = $1^2+2^2+3^2+……….+N^2$

   iii. Sum = N+(N-2)+(N-4)+……..upto 1 or 2 (depending on odd or even).

   iv. Product= 1*2*3…….*N

   v. Sum = 1-2+3-4+5-……N

2. Input any positive integer number ‘N’ which should be less than or equal to 1000. Print the Following :

   Prime number less than N. Where prime Number is a number which is only divisible by itself.

3. Fibonnaci series which should be less than or equal to ‘N’. Where the first two number of the Fibonacci series are 1 and 1.

   The formula for generating Fibonacci series is :

   $F_n=F_{n-1}+F_{n-2}$

4. (i) Input any (1-999999999) digit positive integer number. Print the sum of the digits of the given number.

   (ii) Input any fractional number ≤ 999999999. Print the sum of the digits.

   ex.:- sum of the digits of 123456.78 = 1+2+3+4+5+6+7+8 = 36

5.Write a program which will print those 3 digit numbers where the sum of the cubes of the digits: $1^3+5^3+3^3$-153 .

6.Write a program to display the Following pattern called Floyd’s Triangle.

```
1
2   3
4   5   6
7   8   9   10
11  12  13  14  15
```

7. Input two positive integers<=32767 and print the H .C .F of two number . Allow user to input more numbers and then print the

   H.C.F of all numbers.

8. Calculate e^x where x=.1, .2, .3, .4, .5, .6, .8, .9, .1.0. Verify your result with the standard library function of e^x .

9. Calculate tan(x) = sin(x) / cos(x), using power series of expansion where x = 0 to 360 degree, in step of 5. You have to convert x to radian. Verify your result with the stander library function of tan(x).

10.Calculate the value of PI.

Group B

11. Define 1-dimensional array a[20]. Store 20 distinct numbers in a[20]. Calculate and print the following :
    i) Sum of 20 number(s)
    ii) Average of 20 numbers (avg).
    iii) Maximum number (max).
    iv) Standard deviation (sd). Where

$$sd = \sqrt{\frac{\sum_{i=1}^{20} (a[i] - avg)^2}{20}}$$

12. Write a program to input 20 arbitrary numbers in one-dimensional array. Calculate Frequency of each number.Print the number and its frequency in a tabular form.

13. Write a programs to print the following patterns:

```
(i)   1 0 0 0            (ii)  1 0 0 1            (iii)  1
      0 1 0 0                  0 1 1 0                   1  1
      0 0 1 0                  0 1 1 0                   1  2  1
      0 0 0 1                  1 0 0 1                   1  3  3  1
```

14. Write a program to input numbers in 2-arbitrary arrays such as a[4] ={2,4,5,6} and b[6]={3,4,5,7,10,2}. Create an array c[7]={2,4,5,6,3,7,10} which is the union of array-a and array-b. Keep a provision to vary the size of the arrays a and b. Use proper function to implement the above.

15. Write a program to input numbers in 2-arbitrary arrays such as a[4]={2,4,5,6} and b[6]={3,4,5,7,10,2}. Create an array c[3]={2,4,5} which is the intersection of array-a and array-b. Keep a provision to vary the size of the arrays a and b. Use proper function to implement the above.

16. Given two array A[ ] & B[ ]. Find out the symmetric difference $A \oplus B$
    if $A = \{ 1, 2, 3 \}$
    $B = \{ 2, 8, 6 \}$
    $A \oplus B = \{ 1, 3, 8, 6 \}$

17. Write a program to input two sets such as a[5]={3,5,6,7,0} and b[6]={1,3,7,0,8,9}. Calculate and print:
    (i) Difference of set a[] – set b[] which is c[]={ 5,6 }.
    (ii) Difference of set b[] – set a[] which is d[]={1,8,9}

18. Define 2 dimensional array a (3,3), b(3,3),sum(3,3),diff(3,3),mult(3,3). Store 9 arbitrary numbers in a(3,3) and 9 arbitrary numbers in b(3,3). Do the following:

(i) Calculate sum of a(3,3) and b(3,3) and store in sum(3,3) where sum(i,j)=a(i,j)+b(i,j)
(ii) Calculate difference of a(3,3) and b(3,3) and store in diff(3,3) where diff(i,j)=a(i,j)-b(i,j)
(iii) Calculate product of two arrays a(3,3) and b(3,3) and store in mult(3,3) where mult(i,j)= summation of a(i,k)*b(k,j) over k where k=1 to 3.
Print the result in a tabular form

19.Write a program to convert any decimal number (1-9999999) to any system(2-16).

20. Write a function to calculate n! which is defined as product of 'n' natural numbers and 0! Is defined to be 1.Write another function to calculate the Binomial Coefficient given by,

$$\frac{n!}{k!\,(n-k)!}$$

Group C

21. Write a program to input Roll number (Numeric), Total marks obtained in Final exam of certain number of students.
Do the following :
(i) Sort them according to Total marks in descending order of magnitude.
(ii) Calculate Rank of each student and store in another one-dimensional array.
(iii) Calculate division obtained as below:
If total>=60 then division='1'
If total>=45 then division='2'
If total>=34 then division='3'
If total <34 then division = 'F'
Print Roll number, Total marks, Division and Rank in a tabular form.

22.Input Roll number, Total marks obtained by 'n'(n=1 to 20) students. Write a program to calculate Rank of each student:

23. Write a program which will arrange the positive and negative numbers in a one-dimensional array in such a way that all positive numbers should come first and then all the negative numbers will come without changing original sequence of the numbers.
Example:
Original array contains: 10,-15,1,3,-2,0,-2,-3,2,-9
Modified array: 10,1,3,0,2,-15,-2,-2,-3,-9

24.Write a menu driven program to do the following:

1. Sort numbers using bubble sort
2. Sort numbers using insertion sort
3. Sort numbers using selection sort
4. Sort numbers using 2-way merge sort
5. To exit from the program

Write a program to calculate the time taken by the above sorting methods using time() function. Hints: Total number of elements should not exceed 20,000.

25. Write a program to input maximum 30 numbers in one dimensional array a[30] and their index in another array index[30]. Sort the numbers in ascending order of magnitude.
After that it should display a menu to do the following:

1. To search a number using linear search method

2. To search a number using binary search method
3. To exit from the program

26. Apply recursive call to do the following:
    (i) Input ‘n’(1-200). Calculate sum of ‘n’ numbers.
    (ii) Input ‘n’(1-20). Calculate product of ‘n’ numbers.
    (iii)Input ‘n’(2-20). Print ‘n’ number fibonacci numbers.
    (iv)Input distance ‘n’(1-20). Print how many ways a Robot can move the distance ‘n’ provided the robot can move 1 meter or 2 meters in one step. Extend the idea if the robot can move 1 meter or 2 meter or 3 meter in one step.

27. Input number of disks (1-20). Apply the principle of Towers of Hanoi to transfer those all disks from Peg-1 to Peg-2 using Peg-3. You have to assume that all disks are of different sizes and arranged in ascending order in Peg-1. While transferring those disks you can not change the arrangement of those disks. At the end of the process print how many operations you have performed. Use recursive call in your program.

28. Input any positive integer (n<=3000). Convert the number into Roman numerals.

29. Input any positive integer (n<=9999999). Convert the number into words.

30. Write a program to print three 3-digit numbers such that second number is 2 times as the first number and third number is 3 times as the first number. Note that all 3 numbers should contain 1-9 all digits and there should not be any repetition of the digit.

Group D

Write C program to implement the functions below:

31. int strlen(char s[]) : To return the number of characters in a string s[].

32. i) void strcpy(char s1[], char s2[]) : To copy a string s1[] to s2[].
    ii) void strcat(char s1[], char s2[], char s3[]) : To concatenate s1[] and s2[] to get s3[].

33. i) void strrev(char s[]) : To reverse characters of the original string s[].
    ii) Write a ‘C’ program to test whether a word / sentence is a “Palindrome” or “Not a Palindrome”. Use both recursive and non-recursive methods.

34. i) char tolower(char s) : To convert upper case letter to lower case letter.
    ii) char toupper(char s): To will convert lower case letter to upper case letter.

35. i) void ltrim(char s[]): To eleminate the extra spaces from the left hand side of a string s[ ].
    ii) void rtrim(char s[]): To eleminate the extra spaces from the right hand side of string s[ ].
    iii) void alltrim(char s[]) : To eleminate the extra spaces from the entire string s and will make the spaces uniform throughout the sentence that means between two words there will be only one space and there should not be any leading or trailing spaces in s.

36. i) int nvowel (char s[]) : It will return number of vowels in a sentence s[].
    ii) int nconsonant (char s[]) : It will return number of consonants in a sentence.
    iii) void corcordance(int *nl, int *nv,int *nc, int *ns,int *digit, char s[]): It will return number of letters(nl), number of vowels(nv), number of consonants(nc), number of spaces(ns), number of digits(digit) in a sentence s[ ].

37. i) int strcmp(int n, char s[], char p[]) : It will search a pattern p in s from n-th character of s and will return the first occurence of s in p. if p not found then it will return zero(0). Use this function to calculate how many times a pattern ‘p’ occurs in a string ‘s’.

ii) Write a ‘C’ program that will accept two string as input and compare them to find the length of the greatest common substring between the two.

38. i) int strword(char s[], char w[][]): It will return how many words are there in a sentence s[]. The words to be stored in w[][]. Print the words and the frequency of each word. Finally print how many words are there in s[].

ii) void strnword(char w[]): It will print all possible combination of the letters in a word w. example: if the given word is "GOD" then the output words will be "GOD", "GDO", "OGD","ODG","DOG","DGO". Please note that you can take a word having length = 5. Use both recursive and non-recursive methods.

39. a) Input i) a sentence, ii) a pattern / word to be searched and iii) a new pattern / word to be replaced.

Search the sentence for old pattern / word. If it is found then ask the user “Do you want to modify it (Y/N)?”. If the answer is Y/y then modify the old pattern by the new pattern, otherwise do not modify it. Print the modified sentence.

b) Input i) a sentence, ii) a pattern / word to be searched. Search the sentence for the above pattern. If it is found ask user “Do you want to delete it (Y/N)?”. If the answer is Y/y then delete the pattern, otherwise do not delete it. Print the modified sentence.

40. Write a program to input name, address and telephone number of ‘n’ persons (n<=20). Sort according to the name as a primary key and address as the secondary key. Print the sorted telephone directory.

Group E

41. Write a program to input Roll Number(Numeric), Marks in Paper-1(Out of 100), Marks in Paper-2(Out of 100). Store the data in in an output File : MARKS.DAT.

Read the data from the same file and then do the following:

a. Calculate Total marks,
b. Sort the data according to percentage of marks in descending order of Magnitude.
c. Calculate the Rank of each student those who have obtained >=30 in both the papers.
d. Write Roll number, Marks in Paper-1, Marks in Paper-2, Total marks and Rank to a File : OUTPUT.DAT.

42. Write a program to implement the COPY command of MSDOS so that it will allow user to copy the content of a file to another file. Use command line argument to implement the above problem. Suppose your Program name is FILECOPY.C and FILECOPY.EXE and the name of the source file is MYFILE.TXT and the target file is OUTPUT.TXT then the following command must work:

```
FILECOPY MYFILE.TXT OUTPUT.TXT.
```

Hint: use Windows Operating System

43. Write a program which will split one file into multiple files. Use command line arguments. Apply your program to split movie, image etc.

44. Write a program to display the content of a Text file which means it will behave like TYPE command of MSDOS. Hint: use Windows Operating System

45. Write a program to read a file and transfer the byte number, byte and the ASCII code to another file.

46. Write a program which will copy the Textual matter from any unknown file may be .EXE,.COM, .DLL, .DOC,.BMP, .OBJ,.DBF,.JPG or any other file and the extracted content must be copied to a Target file. Suppose the name of your program file is FILEEXTR.C & FILEEXTR.EXE and the source file is COMMAND.COM and the target file is OUTPUT.TXT then the following command should work:

```
FILEEXTR COMMAND.COM OUTPUT.TXT
```

Hint: use Windows Operating System

46. Write a program to reverse the content of any file and copy the bytes in another file.

47. Write a program to convert all small letters to capital letters in any C-program./ Any text file
48. Write a program to convert all capital letters to small letters in any program. / Any text file.
49. Write a program to remove all comment lines from any C-program.
50. Write a program to extract words from any text file and store in another file. Sort the words in alphabetical order and store them in the same file. Read the sorted file and print the frequency of each word.

Annexure II

CSC-UG-O201

1. Write a program to implement linear and binary search also find the location of its first occurrence
2. Write a program to sort the array in ascending/descending order using a) Quick sort b) Merge sort
3. Write a C Program to perform the basic Matrix operations like add, subtract multiply, transpose, inverse of a matrix
4. Write a program to create a linked list and to perform insert and delete operations (insert at beginning, at last, at any position and same for delete function)
5. Write a program to add two polynomials using a linked list.
6. Write a program to perform insert and delete operations in a circular linked list.
7. Write a program to perform operations on a stack (linked list and array implementation)
8. Write a recursive program to a) find factorial of a given number b) generate first N terms of a fibonacci sequence c) GCD of three numbers.
9. Write a program to perform operations on a circular queue (linked list implementation).
10. Write a program to convert the given infix expression into its postfix form.
11. Write a program to evaluate the postfix expression with a set of values.
12. Write a menu driven program to create binary tree and to perform insert and delete operations.
13. Write a menu driven program to create a binary search tree and to perform inorder, preorder and postorder traversals (recursive and non-recursive)
14. Write a program sort the array of N elements using Heap Sort.
15. Write a program to implement sorting and searching techniques.

Annexure III

CSC-UG-C401

1. Write a program to display names, roll no’s, and grades of 3 students who have appeared in the examination. Declare the class of name, roll no’s and grade. Create an array of class objects. Read and display the contents of the array.
2. Write a program to declare *struct*. Initialize and display contents of member variables.
3. Write a program to declare a class. Declare pointer to class. Initialize and display the contents of the class member.
4. Write a C++ to illustrate the concepts of console I/O operations.
5. Write a C++ program to use scope resolution operator. Display
the various values of the same variables declared at different scope levels.
6. Write a C++ program to allocate memory using new operator.
7.Write a C++ program to create multilevel inheritance. Create classes A1,A2, A3.
8. Write a C++ program to create an array of pointers. Invoke functions
using array objects.
9.Write a C++ program to use pointer for both base and derived classes and call the member function.
10. Write a C++ program to overload unary operator using friend function.
11. Write a C++ program to overload – operator.
12. Write a C++ program to invoke Constructor and Destructor.

Annexure IV

CSC-UG-C401

1. Write a Java Program to define a class, describe its constructor, overload the Constructors and instantiate its object.
2. Write a Java Program to define a class, define instance methods and overload them and use them for dynamic method invocation.
3. Write a Java Program to demonstrate use of sub class.
4. Write a Java Program to demonstrate use of nested class.
5. Write a Java Program to implement array of objects.
6. Write a Java program to practice using String class and its methods.
7. Write a Java program to practice using String Buffer class and its methods.
8. Write a Java Program to implement Vector class and its methods.
9. Write a Java Program to implement Wrapper classes and their methods.
10. Write a Java Program to implement inheritance and demonstrate use of method overriding.
11. Write a Java Program to implement multilevel inheritance by applying various access controls to its data members and methods.
12. Write a program to demonstrate use of implementing interfaces.
13. Write a program to demonstrate use of extending interfaces.
14. Write a Java program to implement the concept of importing classes from user defined package and creating packages.
15. Write a program to implement the concept of threading by extending Thread Class
16. Write a program to implement the concept of threading by implementing Runnable Interface
17. Write a program to implement the concept of Exception Handling using predefined exception.
18. Write a program to implement the concept of Exception Handling by creating user defined exceptions.
19. Write a program using Applet to display a message in the Applet.
20. Write programs for using Graphics class - to display basic shapes and fill them, - draw different items using basic shapes, - set background and foreground colors.

Annexure V

CSC-UG-C502

Exercise 1

Title of the Exercise: Data Definition Language (DDL) Commands

Q1. Create a table called EMP with the following structure.

Name	Type
EMPNO	NUMBER (6)
ENAME	VARCHAR2 (20)
JOB	VARCHAR2 (10)
DEPTNO	NUMBER (3)
SAL	NUMBER (7, 2)

Allow NULL for all columns except ename and job.

Q2: Add a column experience to the EMP table.

Q3: Modify the column width of the job field of EMP table.

Q4: Create dept table with the following structure.

Name	Type
DEPTNO	NUMBER (2)
DNAME	VARCHAR2 (10)
LOC	VARCHAR2 (10)

DEPTNO as the primary key

Q5: create the EMP1 table with ename and empno; add constraints to check the empno value while entering (i.e) empno > 100.

Q6: Drop a column experience from the EMP table.

Q7: Truncate the EMP table and drop the dept table

Exercise 2

Title of the Exercise: Data Manipulation Language (DML) Commands

Q1: Insert a single record into dept table.

Q2: Insert more than a record into EMP table using a single insert command.

Update the EMP table to set the salary of all employees to Rs15000/- who are working as ASP.

Q4: Create a pseudo table employee with the same structure as the table EMP and insert rows into the table using select clauses.

Q5: select employee name, job from the EMP table

Q6: Delete only those who are working as lecturer

Q7: List the records in the EMP table order by salary in ascending order.

Q7: List the records in the EMP table order by salary in ascending order.

Q8: List the records in the EMP table order by salary in descending order.

Q9: Display only those employees whose deptno is 30.

Q10: Display deptno from the table employee avoiding the duplicated values.

Exercise 3

Title of the Exercise: Data Control Language (DCL), Transaction Control Langu/Age (TCL) Commands

Q1: Develop a query to grant all privileges of employees table into departments table.

Q2: Develop a query to grant some privileges of employees table into departments table.

Q3: Develop a query to revoke all privileges of employees table from departments table.
Q4: Develop a query to revoke some privileges of employees table from departments table.
Q5: Write a query to implement the save point.
Q6: Write a query to implement the rollback.
Q6: Write a query to implement the commit.

Exercise 4
Title of the Exercise: In Built Functions
Q1: Display all the details of the records whose employee name starts with ‘A‘.
Q2: Display all the details of the records whose employee name does not starts with ‘A‘.
Q3: Display the rows whose salary ranges from 15000 to 30000.
Q4: Calculate the total and average salary amount of the EMP table.
Q5: Count the total records in the EMP table.
Q6: Determine the max and min salary and rename the column as max_salary and min_salary.
Q7: Display the month between ―1-jun-10‖and 1-aug-10 in full.
Q8: Display the last day of that month in ―05-Oct-09‖.
Q9: Find how many job titles are available in employee table.
Q10: What is the difference between maximum and minimum salaries of employees in the organization?

Exercise 5
Title of the Exercise: Nested Queries and Join Queries
Q1: Display all employee names and salary whose salary is greater than minimum salary of the company and job title starts with ‘M‘.
Q2: Issue a query to find all the employees who work in the same job as Arjun.
Q3: Issue a query to display information about employees who earn more than any employee in dept 1.
Q4: Display the employee details, departments that the departments are same in both the EMP and dept. [EQUIJOIN]
Q5: Display the employee details, departments that the departments are not same in both the EMP and dept. [NON-EQUIJOIN]
Q6: Display the Student name and grade by implementing a left outer join.
Q7: Display the Student name, register no, and result by implementing a right outer join.
Q8: Display the Student name register no by implementing a full outer join.
Q9: Write a query to display their employee names. [Self-Join]
Q10: Display the details of those who draw the salary greater than the average salary.

Annexure VI

CSC-UG-C601

1. Software Reliability and Quality Assurance: Reliability Metric-Musa’s Basic Model.
2. Software Quality Assurance: ISO 9000 and SEI CMM and their Comparison.
3. Software Maintenance: Maintenance Process Models and Reverse Engineering, Estimation of Maintenance Costs.
4. Case Tools and Introduction to UML
5. Each student should be given about 3-4 assignments on SRS, design, testing and allied problems. Different students should be asked to use different tools.
6. The following experiments may be performed for better understanding among students: Case studies shall be provided (designing using UML) based on all the topics covered; Familiarization with CASE (Rational Rose, UML draw, E-Draw etc.) tools.