Computer Science
Syllabus for **B.Sc.**

(As per UGC CBCS w.e.f 2016-17)

Department of Computer Science

Palamuru University
Mahabubnagar
Telangana

Contents

Syllabus for Computer Science

Proposed scheme for **B.Sc.** Programme under **Choice Based Credit System**

Code	Course Title	Course Type	HpW	Credits
	SEMESTER – I			
BS106	Programming in C	DSC–3A	4T+2P=6	4 + 1 =5
	SEMESTER – II			
BS206	Programming in C++	DSC–3B	4T+2P=6	4 + 1 =5
	SEMESTER – III			
BS301	A: SciLab – 1	SEC–1	2T	2
	B: Boolean Algebra			
BS306	Data Structures	DSC–3C	4T+2P=6	4 + 1 =5
	SEMESTER – IV			
BS401	C: SciLab – 2	SEC–2	2T	2
	D: Digital Logic			
BS406	Database Management Systems	DSC–3D	4T+2P=6	4 + 1 =5
	SEMESTER – V			
BS501	Information Technologies –1	GE–1	2	2
BS502	E: Python – 1	SEC–3	2	2
	F: Computer Organization			
BS505	Programming in Java	DSC–3E	3T+2P=5	3 + 1 =4
BS506	Elective–A: Operating Systems	DSE–1E	3T+2P=5	3 + 1 =4
	Elective–B: Software Engineering	DSE–2E		
	SEMESTER – VI			
BS601	Information Technologies –2	GE–2	2T	2
BS602	G: Python – 2	SEC–4	2T	2
	H: Information Security			
BS605	Computer Networks	DSC–3F	3T+2P=5	3 + 1 =4
BS606	Elective–A: PHP with MySQL	DSE–1F	3T+2P=5	3 + 1 =4
	Elective–B: Web Technologies	DSE–2F		
			Total Number of Credits	**48**

DSC–3A Programming in C BS106

Theory	4 Hours/Week	4 credits
Practical	2 Hours/Week	1 credit

Unit – I

Computer Fundamentals: Introduction of Computers, Classification of Computers, Anatomy of a Computer, Memory Hierarchy, Introduction to OS, Operational Overview of a CPU.
Program Fundamentals: Generation and Classification of Programming Languages, Compiling, Interpreting, Loading, Linking of a Program, Developing Program, Software Development.
Algorithms: Definitions, Different Ways of Stating Algorithms (Step-form, Pseudo-code, Flowchart), Strategy for Designing Algorithms, Structured Programming Concept.
Basics of C: Overview of C, Developing Programs in C, Parts of Simple C Program, Structure of a C Program, Comments, Program Statements, C Tokens, Keywords, Identifiers, Data Types, Variables, Constants, Operators and Expressions, Expression Evaluation–precedence and associativity, Type Conversions.

Unit – II

Input-Output: Non-formatted and Formatted Input and Output Functions, Escape Sequences,
Control Statements: Selection Statements – if, if-else, nested if, nested if-else, comma operator, conditional operator, switch; Iterative Statements–while, for, do-while; Special Control Statement–goto, break, continue, return, exit.
Arrays and Strings: One-dimensional Arrays, Character Arrays, Functions from ctype.h, string.h, Multidimensional Arrays.

Unit – III

Functions: Concept of Function, Using Functions, Call-by-Value Vs Call-by-reference, Passing Arrays to Functions, Score of Variables, Storage Classes, Inline Functions, and Recursion.
Pointers: Introduction, Address of Operator (&), Pointer, Uses of Pointers, Arrays and Pointers, Pointers and Strings, Pointers to Pointers, Array of Pointers, Pointer to Array, Dynamic Memory Allocation.

Unit – IV

User-defined Data Types: Declaring a Structure (Union) and its members, Initialization Structure (Union), Accessing members of a Structure (Union), Array of Structures (Union), Structures verses Unions, Enumeration Types.
Files: Introduction, Using Files in C, Working with Text Files, Working with Binary Files, Files of Records, Random Access to Files of Records, Other File Management Functions.

Text Pradip Dey, Manas Ghosh, *Computer Fundamentals and Programming in C* (2e)

References

Ivor Horton, *Beginning C*
Ashok Kamthane, *Programming in C*
Herbert Schildt, *The Complete Reference C*
Paul Deitel, Harvey Deitel, *C How To Program*
Byron S. Gottfried, *Theory and Problems of Programming with C*
Brian W. Kernighan, Dennis M. Ritchie, *The C Programming Language*
B. A. Forouzan, R. F. Gilberg, *A Structured Programming Approach Using C*

C Lab BS106

Practical 2 Hours/Week 1 credit

1 Write a program to find the largest two (three) numbers using if and conditional operator.

2 Write a program to print the reverse of a given number.

3 Write a program to print the prime number from 2 to n where n is given by user.

4 Write a program to find the roots of a quadratic equation using switch statement.

5 Write a program to print a triangle of stars as follows (take number of lines from user):

```
        *
       * * *
      * * * * *
     * * * * * * *
    * * * * * * * * *
```

6 Write a program to find largest and smallest elements in a given list of numbers.

7 Write a program to find the product of two matrices..

8 Write a program to find the GCD of two numbers using iteration and recursion.

9 Write a program to illustrate use of storage classes.

10 Write a program to demonstrate the call by value and the call by reference concepts.

11 Write a program that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.

12 Write a program to illustrate use of data type enum.

13 Write a program to demonstrate use of string functions string.h header file.

14 Write a program that opens a file and counts the number of characters in a file.

15 Write a program to create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.

16 Write a program that opens an existing text file and copies it to a new text file with all lowercase letters changed to capital letters and all other characters unchanged.

Note : Write the Pseudo Code and draw Flow Chart for the above programs.

Recommended to use Open Source Software: GCC on Linux; DevC++ (or) CodeBlocks on Windows 10.

DSC–3B Programming in C++ BS206

Theory	4 Hours/Week	4 credits
Practical	2 Hours/Week	1 credit

Unit – I

Introduction to C++: Applications, Example Programs, Tokens, Data Types, Operators, Expressions, Control Structures, Arrays, Strings, Pointers, Searching and Sorting Arrays.

Functions: Introduction, Prototype, Passing Data by Value, Reference Variables, Using Reference Variables as Parameters, Inline Functions, Default Arguments, Overloading Functions, Passing Arrays to Functions.

Object Oriented Programming: Procedural and Object-Oriented Programming, Terminology, Benefits, OOP Languages, and OOP Applications.

Unit – II

Classes: Introduction, Defining an Instance of a Class, Why Have Private Members? Separating Class Specification from Implementation, Inline Member Functions, Constructors, Passing Arguments to Constructors, Destructors, Overloading Constructors, Private Member Functions, Arrays of Objects, Instance and Static Members, Friends of Classes, Member-wise Assignment, Copy Constructors, Operator Overloading, Object Conversion, Aggregation.

Unit – III

Inheritance: Introduction, Protected Members and Class Access, Base Class Access Specification, Constructors and Destructors in Base and Derived Classes, Redefining Base Class Functions, Class Hierarchies, Polymorphism and Virtual Member Functions, Abstract Base Classes and Pure Virtual Functions, Multiple Inheritance.

C++ Streams: Stream Classes, Unformatted I/O Operations, Formatted I/O Operations.

Unit – IV

Exceptions: Introduction, Throwing an Exception, Handling an Exception, Object-Oriented Exception Handling with Classes, Multiple Exceptions, Extracting Data from the Exception Class, Re-throwing an Exception, Handling the bad_alloc Exception.

Templates: Function Templates–Introduction, Function Templates with Multiple Type, Overloading with Function Templates, Class Templates – Introduction, Defining Objects of the Class Template, Class Templates and Inheritance, Introduction to the STL.

Text	Tony Gaddis, *Starting out with C++: from control structures through objects* (7e)
References	B. Lippman, *C++ Primer* Bruce Eckel, *Thinking in C++* K.R. Venugopal, *Mastering C++* Herbert Schildt, *C++: The Complete Reference* Bjarne Stroustrup, *The C++ Programming Language* Sourav Sahay, *Object Oriented Programming with C++*

C++ Lab BS206

Practical 2 Hours/Week 1 credit

1 Write a program to.

a. Print the sum of digits of a given number.

b. Check whether the given number is Armstrong or not

c. Print the prime number from 2 to n where n is natural number given.

2 Write a program to find largest and smallest elements in a given list of numbers and sort the given list.

3 Write a program to read the student name, roll no, marks and display the same using class and object.

4 Write a program to implement the dynamic memory allocation and de-allocation using new and delete operators using class and object.

5 Write a program to find area of a rectangle, circle, and square using constructors.

6 Write a program to implement copy constructor.

7 Write a program using friend functions and friend class.

8 Write a program to implement constructors
- Default Constructor, Parameterized Constructor, Copy Constructor
- Define the constructor inside/outside of the class
- Implement all three constructors within a single class as well as use multiple classes(individual classes)

9 Write a program to implement the following concepts using class and object
- Function overloading
- Operator overloading (unary/binary(+ and -))

10 Write a program to demonstrate single inheritance, multilevel inheritance and multiple inheritances.

11 Write a program to implement the overloaded constructors in inheritance.

12 Write a program to implement the polymorphism and the following concepts using class and object.
- Virtual functions
- Pure virtual functions

13 Write a program to implement the virtual concepts for following concepts
- Constructor (not applied)
- Destructor (applied)

14 Write a program to demonstrate static polymorphism using method overloading.

15 Write a program to demonstrate dynamic polymorphism using method overriding and dynamic method dispatch.

16 Write a program to implement the template (generic) concepts
- Without template class and object
- With template class and object

Note : Write the Pseudo Code and draw Flow Chart for the above programs.

Recommended to use Open Source Software: GCC on Linux; DevC++ (or) CodeBlocks on Windows.

SEC–1 [A] **SciLab – 1** **BS301**

Theory 2 Hours/Week 2 credits

Unit – I

Introduction to Scilab – what is scilab, downloading & installing scilab, a quick taste of scilab.
The Scilab Environment – manipulating the command line, working directory, comments, variables in memory, recording sessions, the scilab menu bar, demos.
Scalars & Vectors – introduction, initializing vectors in scilab, mathematical operations on vectors, relational operations on vectors, logical operations on vectors, built-in logical functions.

Unit – II

Scalars & Vectors – elementary mathematical functions, mathematical functions on scalars, complex numbers, trigonometric functions, inverse trigonometric functions, hyperbolic functions.
Matrices – introduction, arithmetic operators for matrices, basic matrix processing.
Polynomials – introduction, creating polynomials, basic polynomial commands, finding roots of polynomial, polynomial arithmetic, miscellaneous polynomial handling.

Text Er. Hema Ramachandran, Dr. Achuthsankar S. Nair, *Computer SCILAB–A Free Software to MATLAB*

References

Digite, *Introduction to Scilab*

Digite, *Optimization in Scilab*

Scilab Enterprises, *Scilab for Very Beginners*

Digite, *Introduction to Discrete Probabilities with Scilab*

Note: Student friendly video lecturers pertaining to this course are available at **http://spoken-tutorial.org/**
Teachers are advised to teach this courses in the computer lab itself, so that the interested students may derive some time to perform few programs their own.

SEC–1 [B]

Boolean Algebra

BS301

Theory 2 Hours/Week 2 credits

Unit – I

Introduction Number Systems and Conversion: Digital Systems and Switching Circuits, Number Systems and Conversion, Binary Arithmetic, Representation of Negative Numbers, Binary Codes.

Boolean Algebra: Basic Operations, Boolean Expressions and Truth Tables, Basic Theorems, Commutative, Associative, Distributive, and DeMorgan's Laws, Simplification Theorems, Multiplying Out and Factoring, Complementing Boolean Expressions.

Unit – II

Boolean Algebra: Multiplying Out and Factoring Expressions, Exclusive-OR and Equivalence Operations, The Consensus Theorem, Algebraic Simplification of Switching Expressions, Proving Validity of an Equation, Programmed Exercises.

Applications of Boolean Algebra Minterm and Maxterm Expansions: Conversion of English Sentences to Boolean Equations: Combinational Logic Design Using a Truth Table, Minterm and Maxterm Expansions, General Minterm and Maxterm Expansions, Examples of Truth Table Construction, Design of Binary Adders and Subtracters.

Text

Charles H. Roth, Jr. and Larry L. Kinney*, Fundamentals of Logic Design (7e)*

References

M. Morris Mano, Michael D. Ciletti, *Digital Design (4e)*

A. Saha and N. Manna, *Digital Principles and Logic Design*

M. Rafiquzzaman, *Fundamentals of Digital Logic and Microcontrollers (6e)*

Elliott Mendelson, *Theory and Problems of Boolean Algebra and Switching Circuit*

M. Morris Mano, Charles R. Kime, Tom Martin, *Logic and Computer Design Fundamentals*

DSC–3C Data Structures BS306

Theory	4 Hours/Week	4 credits
Practical	2 Hours/Week	1 credit

Unit – I

Fundamental Concepts: Introduction to Data Structures, Types of Data Structures, Introduction to Algorithm, Pseudo-code, Flow Chart, Analysis of Algorithms.

Linear Data Structure Using Arrays: 1-D Arrays, 2-D Arrays, N-D Arrays, Memory Representation and Address Calculation of 1-D, 2-D, N-D Arrays, Concept of Ordered List, String Manipulation, Pros and Cons of Arrays.

Stacks: Concept, Primitive Operations, Abstract Data Type, Representation Stacks Using Arrays, Prefix, Infix, Postfix Notations for Arithmetic Expression, Applications of Stacks– Converting Infix Expression to Postfix Expression, Evaluating the Postfix Expression, Checking Well-formed (Nested) Parenthesis, Processing of Function Calls, Reversing a String.

Unit – II

Recursion: Introduction, Recurrence, Use of Stack in Recursion, Variants of Recursion, Execution of Recursive Calls, Recursive Functions, Iteration versus Recursion.

Queues: Concept, Primitive Operations, Abstract Data Type, Representation Queues Using Arrays, Circular Queue, Double-Ended Queue, Applications of Queues.

Linked Lists: Introduction, Concept, Terminology, Primitive Operations-creating, inserting, deleting, traversing, Representation of Linked Lists, Linked List Abstract Data Type, Linked List Variants - Singly Linked List, Doubly Linked List, Linear and Circular Linked List, Representation Stacks and Queues Using Linked Singly Lists, Application of Linked List–Garbage Collection.

Unit – III

Trees: Introduction, Representation of a General Tree, Binary Tree Introduction, Binary Tree Abstract Data Type, Implementation of Binary Trees, Binary Tree Traversals – Preorder, Inorder, Postorder Traversals, Applications of Binary Trees Briefly.

Graphs: Introduction, Graph Abstract Data Type, Representation of Graphs, Graph Traversal – Depth-First Search, Breadth-First Search, Spanning Tree – Prim's Algorithm, Kruskal's Algorithm.

Hashing: Introduction, Hash Functions, Collision Resolution Strategies.

Unit – IV

Searching and Sorting: Sequential (Linear) Search, Binary Search, Bubble Sort, Insertion Sort, Selection Sort, Quick Sort, Merge Sort, and Comparison of Sorting Techniques.

Heaps: Concept, Implementation, Abstract Data Type, Heap Sort.

Text

Varsha H. Patil, *Data Structures Using C++*

References

Nell Dale, *C++ Plus Data Structures*

Seymor Lipschutz, *Data Structures (Revised 1e)*

Adam Drozdek, *Data Structures and Algorithms in C++*

Mark Allen Weiss, *Data structures and Algorithm Analysis in C++ (4e)*

D.S. Malik, C++ *Programming: Program Design Including Data Structures (6e)*

Michael Main, Walter Savitch, *Data Structures and Other Objects Using C++ (4e)*

Michael T. Goodrich, R. Tamassia, David M. Mount, *Data Structures and Algorithms in C++*

Yonghui Wu, Jiande Wang, *Data Structure Practice for Collegiate Programming Contests and Education*

Data Structures Lab BS306

Practical 2 Hours/Week 1 credit

1 Write programs to implement the following using an array: a) Stack ADT b) Queue ADT.

2 Write a program to convert the given infix expression to postfix expression using stack.

3 Write a program to evaluate a postfix expression using stack.

4 Write a program to ensure the parentheses are nested correctly in an arithmetic expression.

5 Write a program to find following using Recursion
a) Factorial of +ve Integer b) nth term of the Fibonacci Sequence c) GCD of two +ve integers

6 Write a program to create a single linked list and write functions to implement the following operations.
a) Insert an element at a specified position
b) Delete a specified element in the list
c) Search for an element and find its position in the list
d) Sort the elements in the list ascending order

7 Write a program to create a double linked list and write functions to implement the following operations.
a) Insert an element at a specified position
b) Delete a specified element in the list
c) Search for an element and find its position in the list
d) Sort the elements in the list ascending order

8 Write a program to create singular circular linked lists and function to implement the following operations.
a) Insert an element at a specified position
b) Delete a specified element in the list
c) Search for an element and find its position in the list

9 Write programs to implement the following using a single linked list:
a) Stack ADT b) Queue ADT.

10 Write a program to implement Binary search technique using Iterative method and Recursive methods.

11 Write a program for sorting the given list numbers in ascending order using the following technique: Bubble sort and Selection sort

12 Write a program for sorting the given list numbers in ascending order using the following technique: Insertion sort and Quick sort

13 Write a program for sorting the given list numbers in ascending order using the following technique: Merge sort and Heap sort

14 Write a program to traverse a binary tree in following way.
a) Pre-order b) In-order c) Post-order

15 Write a program to the implementation graph traversals – BFS and DFS.

16 Write a program to find the minimum spanning tree for a weighted graph using
a) Prim's Algorithm b) Kruskal's Algorithm.

Note : Write the Pseudo Code for the above programs.

Recommended to use Open Source Software: GCC on Linux; DevC++ (or) CodeBlocks on Windows.

SEC–2 [A]

SciLab – 2

BS401

Theory 2 Hours/Week 2 credits

Unit – I

Programming in scilab – introduction, variables & variable names, assignment statements, arithmetic, relational, logical operators, input & output, flow control/branching/conditional statements, break and continue, handling matrices with loops, scripts, the concept of functions, user defined functions, special function commands.

Menus and Dialog Boxes – introduction, a simple menu example, scilab window with greetings menu added, executing submenus from command line, linking menus to scilab code from external files, entering data through dialog boxes, printing a message in a message box, dialog box for entering a matrix.

Unit – II

Graphic Output – introduction, 2d plotting, function versions for graphic commands, 3d plotting, other graphic primitives, other graphic commands.

String Handling Functions – symbolic processing in scilab, creation of a linear combination of arguments, string to ASCII conversion, creation of a string of blank characters, conversion of a string to uppercase and lowercase, string matching, string concatenation, reversing a string, replacement of a string by another, length of a string, type checking.

Statistics – introduction, basic statistical functions, applying statistical functions on matrices, distributions, frequency of values of a matrix or vector, centre, weighted centre, central moment, correlation, covariance, variance matrix, percentiles, frequencies, cumulative sum, difference of two independent samples, fisher test.

Text Er. Hema Ramachandran, Dr. Achuthsankar S. Nair, *Computer SCILAB–A Free Software to MATLAB*

References

Digite, *Introduction to Scilab*

Digite, *Optimization in Scilab*

Scilab Enterprises, *Scilab for Very Beginners*

Digite, *Introduction to Discrete Probabilities with Scilab*

Note: Student friendly video lecturers pertaining to this course are available at **http://spoken-tutorial.org/**

Teachers are advised to teach this courses in the computer lab itself, so that the interested students may derive some time to perform few programs their own.

SEC–2 [B]

Digital Logic

BS401

Theory 2 Hours/Week 2 credits

Unit – I

Karnaugh Maps: Minimum Forms of Switching Functions, Two- and Three-Variable Karnaugh Maps, Four-Variable Karnaugh Maps, Determination of Minimum Expressions Using Essential Prime Implicants, Other Uses of Karnaugh Maps, Other Forms of Karnaugh Maps, Programmed Exercises.

Multi-Level Gate Circuits NAND and NOR Gates: Multi-Level Gate Circuits, NAND and NOR Gates, Design of Two-Level NAND- and NOR-Gate Circuits, Design of Multi-Level NAND- and NOR-Gate Circuits, Circuit Conversion Using Alternative Gate Symbols, Design of Two-Level, Multiple-Output Circuits, Multiple-Output NAND- and NOR-Gate Circuits.

Unit – II

Combinational Circuit Design and Simulation Using Gates: Design of Circuits with Limited Gate Fan-In, Gate Delays and Timing Diagrams, Hazards in Combinational Logic, Simulation and Testing of Logic Circuits.

Multiplexers, Decoders: Multiplexers, Three-State Buffers, Decoders and Encoders, Read-Only Memories.

Text	Charles H. Roth, Jr. and Larry L. Kinney*, Fundamentals of Logic Design (7e)*
References	M. Morris Mano, Michael D. Ciletti, *Digital Design (4e)* A. Saha and N. Manna, *Digital Principles and Logic Design* M. Rafiquzzaman, *Fundamentals of Digital Logic and Microcontrollers (6e)* Elliott Mendelson, *Theory and Problems of Boolean Algebra and Switching Circuit* M. Morris Mano, Charles R. Kime, Tom Martin, *Logic and Computer Design Fundamentals*

DSC–3D Database Management Systems BS406

Theory	4 Hours/Week	4 credits
Practical	2 Hours/Week	1 credit

Unit – I

Introduction to Databases: Introduction, Traditional File-Based Systems, Database Approach, Roles in the Database Environment, Advantages and Disadvantages of DBMSs, The Three-Level ANSI-SPARC Architecture, Database Languages, Data Models, Functions of a DBMS, Components of a DBMS.

Relational Model: Introduction, Terminology, Integrity Constraints, Views.

The Relational Algebra: Unary Operations, Set Operations, Join Operations, Division Operation, Aggregation and Grouping Operations.

Unit – II

SQL: Introduction, Data Manipulation–Simple Queries, Sorting Results, Using the SQL Aggregate Functions, Grouping Results, Sub-queries, ANY and ALL, Multi-table Queries, EXISTS and NOT EXIST, Combining Result Tables, Database Updates.

SQL: The ISO SQL Data Types, Integrity Enhancement Feature–Domain Constraints, Entity Integrity, Referential Integrity, General Constraints, Data Definition–Creating a Database, Creating a Table, Changing a Table Definition, Removing a Table, Creating an Index, Removing an Index, Views–Creating a View, Removing a View, View Resolution, Restrictions on Views, View Updatability, WITH CHECK OPTION, Advantages and Disadvantages of Views, View Materialization, Transactions, Discretionary Access Control–Granting Privileges to Other Users, Revoking Privileges from Users.

Advanced SQL: The SQL Programming Language–Declarations, Assignments, Control Statements, Exceptions, Cursors, Subprograms, Stored Procedures, Functions, and Packages, Triggers, Recursion.

Unit – III

Entity–Relationship Modeling: Entity Types, Relationship Types, Attributes, Keys, Strong and Weak Entity Types, Attributes on Relationships, Structural Constraints, Problems with ER Models–Fan Traps, Chasm Traps.

Enhanced Entity–Relationship Modeling: Specialization/Generalization, Aggregation, Composition.

Functional–Dependencies: Anomalies, Partial Functional Dependency, Transitive Functional Dependency, Multi Valued Dependency, Join Dependency.

Normalization: The Purpose of Normalization, How Normalization Supports Database Design, Data Redundancy and Update Anomalies, Functional Dependencies in brief, The Process of Normalization,1NF, 2NF, 3NF, BCNF. The Database Design Methodology for Relational Databases (Appendix–D).

Unit – IV

Transaction Management: Transaction Support–Properties of Transactions, Database Architecture, Concurrency Control–The Need for Concurrency Control, Serializability and Recoverability, Locking Methods, Deadlock, Time Stamping Methods, Multi-version Timestamp Ordering, Optimistic Techniques, Granularity of Data Items, Database Recovery–The Need for Recovery, Transactions and Recovery, Recovery Facilities, Recovery Techniques, Nested Transaction Model.

Security: Database Security–Threats, Computer-Based Controls–Authorization, Access Controls, Views, Backup and Recovery, Integrity, Encryption, RAID.

Text	Thomas M. Connolly, Carolyn E. Begg, *Database Systems–A Practical Approach to Design, Implementation, and Management (6e)*
References	Sharon Allen, Evan Terry, *Beginning Relational Data Modeling* Jeffrey A. Hoffer, V. Ramesh, Heikki Topi, *Modern Database Management* Raghu Ramakrishnan, Johannes Gehrke, *Database Management Systems* Ramez Elmasri, Shamkant B. Navathe, *Fundamentals of Database Systems* Abraham Silberschatz, Henry F. Korth, S. Sudarshan, *Database System Concepts* C Coronel, S Morris, Peter Rob, *Database Systems: Design, Implementation, and Management*

Database Management Systems Lab BS406

Practical 2 Hours/Week 1 credit

Consider the relational schema for part of the **DreamHome** case study is:

Branch (branchNo, street, city, postcode)
Staff (staffNo, fName, IName, position, sex, DOB, salary, branchNo)
PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo, staffNo, branchNo)
Client (clientNo, fName, IName, telNo, prefType, maxRent, eMail)
PrivateOwner (ownerNo, fName, IName, address, telNo, eMail, password)
Viewing (clientNo, propertyNo, viewDate, comment)
Registration (clientNo, branchNo, staffNo, dateJoined)

1. Create a database with name "DreamHome" and now create all the tables listed above with constraints.
2. Insert a new row into the table supplying data for all columns.
3. Modify data in the database using UPDATE
4. Delete data from the database using DELETE
5. Changing a table definition using ALTER
6. Removing a table using DROP
7. Removing rows in table using TRUNCATE
8. Create an index and removing an index
9. Practice other standard SQL commands for creating, modifying, displaying data of tables.
10. List full details of all staff.
11. List all staff with a salary greater than £10000.
12. List the property numbers of all properties that have been viewed.
13. Produce a list of salaries for all staff, showing only the staffNo, fName, IName, and salary details.
14. List all cities where there is either a branch office or a property for rent.
15. List all cities where there is a branch office but no properties for rent.
16. List all cities where there is both a branch office and at least one property for rent.
17. List the names and comments of all clients who have viewed a property for rent.
18. Produce a status report on property viewings.
19. List complete details of all staff who work at the branch in Glasgow.
20. List the addresses of all branch offices in London or Glasgow
21. List all staff with a salary between £20,000 and £30,000.
22. Identify all clients who have viewed all properties with three rooms.
23. How many properties cost more than £350 per month to rent?
24. How many different properties were viewed in May 2013?
25. Find the total number of Managers and the sum of their salaries.
26. Find the minimum, maximum, and average staff salary.
27. Find the number of staff working in each branch and the sum of their salaries.
28. List all managers and supervisors.
29. Find all owners with the string 'Glasgow' in their address.
30. List the details of all viewings on property PG4 where a comment has not been supplied.
31. Produce a list of salaries for all staff, arranged in descending order of salary.
32. Produce an abbreviated list of properties arranged in order of property type.
33. Find the number of staff working in each branch and the sum of their salaries.
34. For each branch office with more than one member of staff, find the number of staff working in each branch and the sum of their salaries.
35. List the staff who work in the branch at '163 Main St'.
36. List all staff whose salary is greater than the average salary, and show by how much their salary is greater than the average.
37. List the properties that are handled by staff who work in the branch at '163 Main St'.
38. Find all staff whose salary is larger than the salary of at least one member of staff at branch B003.
39. Find all staff whose salary is larger than the salary of every member of staff at branch B003
40. List the names of all clients who have viewed a property, along with any comments supplied.
41. For each branch office, list the staff numbers and names of staff who manage properties and the properties that they manage.
42. For each branch, list the staff numbers and names of staff who manage properties, including the city in which the branch is located and the properties that the staff manage.

43. Find the number of properties handled by each staff member, along with the branch number of the member of staff.
44. List all branch offices and any properties that are in the same city.
45. List all properties and any branch offices that are in the same city.
46. List the branch offices and properties that are in the same city along with any unmatched branches or properties.
47. Find all staff who work in a London branch office.
48. Construct a list of all cities where there is either a branch office or a property.
49. Construct a list of all cities where there is both a branch office and a property.
50. Create a view so that the manager at branch B003 can see the details only for staff who work in his or her branch office.
51. Create a view of the staff details at branch B003 that excludes salary information, so that only managers can access the salary details for staff who work at their branch.
52. Create a view of staff who manage properties for rent, which includes the branch number they work at, their staff number, and the number of properties they manage.
53. Removing a view using DROP VIEW
54. Give the user with authorization identifier Manager all privileges on the Staff table.
55. Give users Personnel and Director the privileges SELECT and UPDATE on column salary of the Staff table.
56. Revoke the privilege SELECT on the Branch table from all users.
57. Revoke all privileges you have given to Director on the Staff table.
58. Demonstrate exceptions in PL/SQL
59. Demonstrate cursors in PL/SQL
60. Write PL/SQL queries to create procedures.
61. Write PL/SQL queries to create functions.
62. Write PL/SQL queries to create package.
63. Write PL/SQL queries to create triggers.
64. Write PL/SQL queries using recursion.

Consider the relational schema for part of the **Hotel** case study is:

Hotel (<u>hotelNo</u>, hotelName, city)
Room (<u>roomNo</u>, <u>hotelNo</u>, type, price)
Booking (<u>hotelNo</u>, <u>guestNo</u>, <u>dateFrom</u>, dateTo, roomNo)
Guest (<u>guestNo</u>, guestName, guestAddress)

65. Create a database with name “Hotel” and now create all the tables listed above with constraints.
66. Insert a new row into the table supplying data for all columns.
67. Modify data in the database using UPDATE
68. Delete data from the database using DELETE
69. Changing a table definition using ALTER
70. Removing a table using DROP
71. Removing rows in table using TRUNCATE
72. Practice other standard SQL commands for creating, modifying, displaying data of tables.
73. List full details of all hotels.
74. List full details of all hotels in London.
75. List the names and addresses of all guests living in London, alphabetically ordered by name.
76. List all double or family rooms with a price below £40.00 per night, in ascending order of price.
77. List the bookings for which no dateTo has been specified.
78. How many hotels are there?
79. What is the average price of a room?
80. What is the total revenue per night from all double rooms?
81. How many different guests have made bookings for August?
82. List the price and type of all rooms at the Grosvenor Hotel.
83. List all guests currently staying at the Grosvenor Hotel.
84. List the details of all rooms at the Grosvenor Hotel, including the name of the guest staying in the room.
85. What is the total income from bookings for the Grosvenor Hotel today?
86. List the rooms that are currently unoccupied at the Grosvenor Hotel.
87. What is the lost income from unoccupied rooms at the Grosvenor Hotel?
88. List the number of rooms in each hotel.
89. List the number of rooms in each hotel in London.
90. What is the average number of bookings for each hotel in August?
91. What is the most commonly booked room type for each hotel in London?
92. What is the lost income from unoccupied rooms at each hotel today?
93. Insert rows into each of these tables.

94. Update the price of all rooms by 5%.
95. Demonstrate that queries written using the UNION operator and same can be rewritten using the OR.
96. Apply the syntax for inserting data into a table.
97. Create a view containing the cheapest hotels in the world.
98. Create the Hotel table using the integrity enhancement features of SQL.
99. Create a database trigger for the following situations:
    (a) The price of all double rooms must be greater than £100.
    (b) The price of double rooms must be greater than the price of the highest single room.
    (c) A booking cannot be for a hotel room that is already booked for any of the specified dates.
    (d) A guest cannot make two bookings with overlapping dates.
    (e) Maintain an audit table with the names and addresses of all guests who make bookings for hotels in London (do not store duplicate guest details).

Given relation schemas are

Sailors(sid : integer, sname : string, rating : integer, age : real)
Boats(bid : integer, bname : string, color : string)
Reserves(sid : integer , bid : integer, day : date)

100. Find the names and ages of all sailors.
101. Find all sailors with a rating above 7.
102. Find the names of sailors who have reserved boat 103.
103. Find the sids of sailors who have reserved a red boat.
104. Find the names of sailors who have reserved a red boat.
105. Find the colors of boats reserved by Lubber.
106. Find the names of sailors who have reserved at least one boat.
107. Find the names of sailors who have reserved at least two boats.
108. Compute increments for the ratings of persons who have sailed two different boats on the same day.
109. Find the ages of sailors whose name begins and ends with B and has at least three characters.
110. Find the names of sailors who have reserved a red or a green boat.
111. Find the names of sailors who have reserved a red and a green boat.
112. Find the sids of all sailors who have reserved red boats but not green boats.
113. Find all sids of sailors who have a rating of 10 or have reserved boat 104.
114. Find the names of sailors who have not reserved a red boat.
115. Find sailors whose rating is better than some sailor called Horatio.
116. Find sailors whose rating is better than every sailor called Horatio.
117. Find the names of sailors who have reserved all boats.
118. Find the names of sailors who have reserved at least two boats.
119. Find the names of sailors who have reserved all boats called Interlake.
120. Find sailors who have reserved all red boats.
121. Find the sailor name, boat id, and reservation date for each reservation.
122. Find the sids of sailors with age over 20 who have not reserved a red boat.
123. Find the average age of all sailors.
124. Find the average age of sailors with a rating of 10.
125. Find the name and age of the oldest sailor.
126. Count the number of different sailor names.
127. Find the names of sailors who are older than the oldest sailor with a rating of 10.
128. Find the sailors with the highest rating.
129. Find the age of the youngest sailor for each rating level.
130. Find age of the youngest sailor who is eligible to vote for each rating level with at least 2 such sailors.
131. Find the average age of sailors for each rating level that has at least two sailors.
132. For each red boat, find the number of reservations for this boat.
133. Find the average age of sailors who are of voting age (i.e., at least 18 years old) for each rating level that has at least two sailors.
134. Delete the records of sailors who have rating 8 (deleting some rows in a table).
135. Loading data which is present in the text into the table.

Note : Recommended to use open source database software like MySQl, MongoDB, PostgreSQL, etc…
In practical examination, students have to

- Create database
- Create tables with their integrity constraints.
- Insert the data into tables and then execute the queries.
- Answer any **six** queries from **ten** queries given by the examiner.

GE–1 Information Technologies – 1 BS501

Theory 2 Hours/Week 2 credits

Unit – I

Introduction to Computers: What is a Computer? Characteristics of Computers, Generations of Computers, Classification of Computers, Basic Computer Organization, Applications of Computers.
Input and Output Devices: Input Devices, Output Devices, Soft Copy Devices, Hard Copy Devices.
Computer Memory and Processors: introduction, Memory Hierarchy, Processor Registers, Cache Memory, Primary Memory, Secondary Storage Devices, Hard Disks, Optical Drives, USB Flash Drives, Memory Cards.

Unit – II

Computer Software: Introduction, Classification of Computer Software, System Software, Applications Software, Firmware, Middleware, Acquiring Computer Software.
Operating Systems: Introduction, Evolution of OS, Process Management, Memory Management, File Management, Device Management, Security Management, Command Interpreter, Windows, Linux.

Text Reema Thareja, *Fundamentals of Computers*

References

P. K. sinha, *Computer Fundamentals*
Anita Goel, *Computer Fundamentals*
V. Rajaraman, *Fundamentals of Computers*
E. Balagurusamy, *Fundamentals of Computers*
J. Glenn Brookshear, Dennis Brylow, *Computer Science An Overview*

Note: Student friendly video lecturers pertaining to this course are available at **http://spoken-tutorial.org/**
Teachers are advised to teach this courses in the computer lab itself, so that the interested students may derive some time to perform few programs their own.

SEC–3 [A]

Python – 1

BS502

Theory 2 Hours/Week 2 credits

Unit – I

Introduction to Python: Python, Features of Python, Execution of a Python Program, Viewing the Byte Code, Flavors of Python, Python Virtual Machine, Frozen Binaries, Memory Management in Python, Garbage Collection in Python, Comparisons between C and Python, Comparisons between Java and Python.
Writing Our First Python Program: Installing Python for Windows, Installing numpy, Setting the Path to Python, Writing Our First Python Program, Executing a Python Program, Getting Help in Python, Getting Python Documentation Help, Reopening the Python Program in IDLE.
Data types in Python: Comments in Python, Doc strings, How Python Sees Variables, Data types in Python, Built-in data types, bool Data type, Sequences in Python, Sets, Literals in Python, Determining the Data type of a Variable, What about Characters, User-defined Data types, Constants in Python, Identifiers and Reserved words, Naming Conventions in Python.

Unit – II

Operators in Python: Arithmetic Operators, Assignment Operators, Unary Minus Operator, Relational Operators, Logical Operators, Boolean Operators, Bitwise Operators, Membership Operators, Identity Operators, Operator Precedence and Associativity, Mathematical Functions.
Input and Output: Output statements, Input Statements, Command Line Arguments.
Control Statements: Control Statements, The if Statement, A Word on Indentation, The if … else Statement, The if … elif … else Statement, The while Loop, The for Loop, Infinite Loops, Nested Loops, The else Suite, The break Statement, The continue Statement, The pass Statement, The assert Statement, The return Statement.

Text R. Nageswara Rao, *Corer Python Programming,* Dreamtech Press

Reference s

Mark Lutz, *Learning Python*
Tony Gaddis, *Starting Out With Python*
Kenneth A. Lambert, *Fundamentals of Python*
James Payne, *Beginning Python using Python 2.6 and Python 3*
Paul Gries, *Practical Programming: An Introduction to Computer Science using Python 3*

Note: Student friendly video lecturers pertaining to this course are available at
http://spoken-tutorial.org/
Teachers are advised to teach this courses in the computer lab itself, so that the interested students may derive some time to perform few programs their own.

SEC–3 [B]

Computer Organization

BS502

Theory 2 Hours/Week 2 credits

Unit – I

Latches and Flip-Flops: Introduction, Set-Reset Latch, Gated Latches, Edge-Triggered D Flip-Flop, S-R Flip-Flop, J-K Flip-Flop, T Flip-Flop, Flip-Flops with Additional Inputs, Asynchronous Sequential Circuits.

Registers and Counters: Registers and Register Transfers, Shift Registers, Design of Binary Counters, Counter Design Using D Flip-Flops, Counter Design Using S-R and J-K Flip-Flops, Derivation of Flip-Flop Input Equations.

Unit – II

Sequential Circuit Design: Design Example—Code Converter, Design of Iterative Circuits, Design of Sequential Circuits Using ROMs and PLAs, Simulation and Testing of Sequential Circuits, Computer-Aided Design.

Circuits for Arithmetic Operations: Serial Adder with Accumulator, Design of a Binary Multiplier, Design of a Binary Divider.

Text

Charles H. Roth, Jr. and Larry L. Kinney*, Fundamentals of Logic Design (7e)*

References

M. Morris Mano, Michael D. Ciletti, *Digital Design (4e)*

A. Saha and N. Manna, *Digital Principles and Logic Design*

M. Rafiquzzaman, *Fundamentals of Digital Logic and Microcontrollers (6e)*

Elliott Mendelson, *Theory and Problems of Boolean Algebra and Switching Circuit*

M. Morris Mano, Charles R. Kime, Tom Martin, *Logic and Computer Design Fundamentals*

DSC–3E Programming in Java BS505

Theory	3 Hours/Week	3 credits
Practical	2 Hours/Week	1 credit

Unit – I

Introduction: Java Essentials, JVM, Java Features, Creation and Execution of Programs, Data Types, Type Conversion, Casting, Conditional Statements, Loops, Branching Mechanism, Classes, Objects, Class Declaration, Creating Objects, Method Declaration and Invocation, Method Overloading, Constructors–Parameterized Constructors, Constructor Overloading, Cleaning-up unused Objects, Class Variables & Methods-static Keyword, this Keyword, One-Dimensional Arrays, Two-Dimensional Arrays, Command-Line Arguments, Inner Class.

Inheritance: Introduction, Types of Inheritance, extends Keyword, Examples, Method Overriding, super, final Keywords, Abstract classes, Interfaces, Abstract Classes Verses Interfaces.

Unit – II

Packages–Creating and Using Packages, Access Protection, Wrapper Classes, String Class, StringBuffer Class.

Exception: Introduction, Types, Exception Handling Techniques, User-Defined Exception.

Multithreading: Introduction, Main Thread, Creation of New Threads – By Inheriting the Thread Class or Implementing the Runnable Interface, Thread Lifecycle, Thread Priority, Synchronization.

Input/Output: Introduction, java.io Package, File Class, FileInputStream Class, FileOutputStream Class, Scanner Class, BufferedInputStream Class, BufferedOutputStream Class, RandomAccessFile Class.

Unit – III

Applets: Introduction, Example, Life Cycle, Applet Class, Common Methods Used in Displaying the Output. Event Handling: Introduction, Types of Events, Example. AWT: Introduction, Components, Containers, Button, Label, Checkbox, Radio Buttons, Container Class, Layouts. Swing: Introduction, Differences between Swing and AWT, Jframe, Japplet, Jpanel, Components in Swings, Layout Managers, Jtable, Dialog Box.

Database Handling Using JDBC: Introduction, Types of JDBC Drivers, Load the Driver, Establish Connection, Create Statement, Execute Query, Iterate Resultset, Scrollable Resultset, Developing a JDBS Application.

Text Sachin Malhotra, Saurabh Choudhary, *Programming in Java (2e)*

References

Bruce Eckel, *Thinking in Java (4e)*

Herbert Schildt, *Java: The Complete Reference (9e)*

Y. Daniel Liang, *Introduction to Java Programming (10e)*

Paul Deitel, Harvey Deitel, *Java: How To Program (10e)*

Cay S. Horsttnann*, Core Java Volume I – Fundamentals (10e)*

C. Thomas Wu, *An introduction to object-oriented programming with Java (5e)*

Tony Gaddis, *Starting Out with Java From Control Structures Through Objects (6e)*

Jeanne Boyarsky, Scott Selikoff, *OCA: Oracle Certified Associate Java SE 8 Programmer–I Study Guide*

Java Lab BS505

Practical 2 Hours/Week 1 credit

1 Write java programs to find the following
a) largest of given three numbers b) reverses the digits of a number
c) given number is prime or not d) GCD of given two integers

2 Write java programs that implement the following
a) default constructor b) parameterized constructor c) constructor overloading

3 a) Write a java program to find the smallest of given list integers using array and scanner class.
b) Write a java program for multiplication of two matrices.

4 a) Write a java program for demonstrating an inner classes or nested classes.
b) Write a java program to implement method overloading, method overriding, dynamic method dispatch

5 Write a java program to implement single, multilevel, hierarchal, multiple, hybrid inheritances.

6 Write java programs that demonstrate the use of abstract, this, super, static, final keywords

7 a) Write a java program for creating a package and using a package.
b) Write a java program to demonstrate the use of wrapper classes.

8 a) Write a java program using all five keywords of exception handling mechanism.
b) Write a java program for creating customized (user) exception

9 a) Write a java program that checks whether a given string is a palindrome or not.
b) Write a java program for sorting a given list of names in ascending order.

10 a) Write a java program to create a file, write the data and display the data.
b) Write a java program that reads a file name from user and displays its information.

11 a) Write a java program for controlling main thread.
b) Write a java program for creating new thread by extending Thread class.

12 a) Write a java program for creating new thread by implementing Runnable interface.
b) Write a java program for thread synchronization.

13 a) Write a java program to create following AWT components: Button, Checkbox, Choice, and List.
b) Write java programs to create AWT application using containers and layouts.

14 a) Write java programs to create a simple Applet and create swing based Applet.
b) Write a java program to handle different types of events in a swing application.

15 Write java programs to create a swing application using swing components and layouts.

16 Write a java program to store and retrieve data from database using JDBC.

Note : Write the program using simple text editors (not IDE), compile and run from command prompt.

Encourage students to develop small java applications using IDE, like giving as assignment.

Write a small java application using some features of java.

DSE–1E Operating Systems BS506

Theory	3 Hours/Week	3 credits
Practical	2 Hours/Week	1 credit

Unit – I

Introduction: Computer-System Architecture, Computing Environments.
Operating-System Structures: Operating-System Services, User Interface for Operating-System, System Calls, Types of System Calls, Operating System Structure.
Process Management: Process Concept, Process Scheduling, Operations on Processes, Inter process Communication, Examples–Producer-Consumer Problem.
Process Synchronization: Critical-Section Problem, Peterson's Solution, Synchronization, Semaphores, Monitors.

Unit – II

CPU Scheduling: Concepts, Scheduling Criteria, Scheduling Algorithms.
Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.

Unit – III

Main Memory: Introduction, Swapping, Contiguous Memory Allocation, Segmentation, Paging.
Virtual Memory: Introduction, Demand Paging, Page Replacement, Allocation of Frames, Thrashing.
Mass-Storage Structure: Overview, Disk Scheduling, RAID Structure.
File Systems: File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, Protection.
File System Implementation, Directory Implementation, Allocation Methods, Free-Space Management.

Text

Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, *Operating System Concepts (9e)*

References

Naresh Chauhan, *Principles of Operating Systems*
Thomas W. Doeppner, *Operating Systems in Depth*
Andrew S. Tanenbaum, *Modern Operating Systems*
William Stallings, *Operating Systems – Internals and Design Principles*
Dhananjay M. Dhandhere, *Operating Systems – A Concept Based Approach*

Operating Systems Lab BS506

Practical 2 Hours/Week 1 credit

1 a) Use vi editor to create different files, writing data into files, modifying data in files.
b) Use different types of Unix commands on the files created in first program.

2 Write shell programs using ‘case’, ‘then’ and ‘if’ & ’else’ statements.

3 Write shell programs using while, do-while and for loop statements.

4 a) Write a shell script that accepts two integers as its arguments and computers the value of first number raised to the power of the second number.
b) Write a shell script that takes a command –line argument and reports on whether it is directory, a file, or something else.

5 a) Write a shell script that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers..
b) Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.

6 a) Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
b) Develop an interactive script that ask for a word and a file name and then tells how many times that word occurred in the file.

7 Write a program that simulate the following Unix commands like ls, mv, cp.

8 Write a program to convert upper case to lower case letters of a given ASCII file.

9 Write a program to program to search the given pattern in a file.

10 Write a program to demonstrate FCFS process schedules on the given data.

11 Write a program to demonstrate SJF process schedules on the given data.

12 Write a program to demonstrate Priority Scheduling on the given burst time and arrival times.

13 Write a program to demonstrate Round Robin Scheduling on the given burst time and arrival times.

14 Write a program to implementing Producer and Consumer problem using Semaphores.

15 Write a program to simulate FIFO, LRU, LFU Page replacement algorithms.

16 Write a program to simulate Sequential, Indexed, and Linked file allocation strategies.

Note : Recommended to use Open Source Software like Fedora, Ubuntu, CentOS, etc...

DSE–2E Software Engineering BS506

Theory	3 Hours/Week	3 credits
Practical	2 Hours/Week	1 credit

Unit – I

Software Engineering – Introduction, Program Versus Software, Software Engineering, Software Development Process and its Stages, Generic Software Development Process Models, Code of Ethics and Professional Practice, Software Development and Maintenance Cost Breakup.

Requirement Engineering Processes – Requirement Engineering Process, Feasibility Study, Cost and Benefit Analysis, Requirement Specification, Characteristics of a Good Requirement and Validation Techniques, Requirements Management Planning, Process of Requirement Change Management.

Software Requirement Specifications – Introduction, Stakeholder Analysis, Software Requirements Document, IEEE Standard of Software Requirement Specifications, Organizing Functional Requirements, Traceability and Validation of Specifications.

Unit – II

Architectural Styles – Introduction, Architecture Styles, Object Oriented Architecture, Inter-organizational Communication, Cloud Computing Architecture Style, Core, Configurable and Customizable Architecture, Design Models, Architectural Design Principles.

Object Oriented System Analysis – Introduction, Object Oriented Design, Object Oriented Design Models, Object Oriented Analysis, Data Modeling, Comparison Between Top Down Structured and Object Oriented Analysis, Description of Logical and Static Modeling, Identification of Class Relationships.

Object Oriented Design Using UML – Introduction, Sequence Diagram, State Machine Diagram, Timing Diagram, Describing Detailed Object Oriented Design, Decision Tree and Decision Table, Composite Structure Diagram, Generating Test Cases, Moving Towards Physical Design, Structured Methods.

Unit – III

Software Development – Introduction, Good Coding Practices, Code Reuse, Design Pattern, Generator Based Reuse, Application/Software Developed on Product Lines Approach, Component Based Software Engineering, Agile Methods.

Verification, Validation and Software Testing – Introduction, Software Verification and Validation Process, Software Testing, System Testing, Object Oriented Testing Strategy, Test Cases, Equivalence Partitioning (Black Box Testing), Art of Debugging.

Measurement and Metrics for Assessing Software Quality – Introduction, ISO 9126 Quality Standards, Quality Management Models, Ways to Build Quality in Software, Software Quality Control and Metrics, Defect Density Metrics, Chidamber and Kemerer Metric Suites for Object Oriented System, Class Coupling Metric-Coupling Between Objects, Monitoring Dynamic Quality Attributes (Visible Externally) of a Software.

Text

Rajesh Narang, *Software Engineering: Principles and Practices*

References

Ian Sommerville, *Software Engineering*

R. Mall, *Fundamentals of Software Engineering*

Pankaj Jalote, An Integrated *Approach to Software Engineering*

Frank Tsui, Orlando Karam, Barbara Bernal, *Essentials of Software Engineering*

Roger S Pressman, B R Maxim, *Software Engineering – A Practitioner's Approach*

Grady Booch, *The Unified Modeling Language User Guide*

Software Engineering Lab BS506

Practical 2 Hours/Week 1 credit

Case Studies:

1 Banking System

2 Hotel management system

3 Inventory Control System

4 Library management system

5 Railway Reservation System

Choose any two of above case studies and do the following exercises for that case studies

1 Write the complete problem statement

2 Write the software requirements specification document

3 Draw the entity relationship diagram

4 Draw the data flow diagrams

5 Draw use case diagrams

6 Draw activity diagrams for all use cases

7 Draw sequence diagrams for all use cases

8 Draw collaboration diagram

9 Assign objects in sequence diagrams to classes and make class diagram.

Note : To draw dataflow diagrams using Microsoft Visio Software, SmartDraw, etc…

To draw UML diagrams using Rational Rose Software, StarUML, etc...

The teacher should define the boundaries for the above case study problems and make the practice of problems mentioned.

GE–2 Information Technologies – 2 BS601

Theory 2 Hours/Week 2 credits

Unit – I

Introduction to Algorithms and Programming Languages: Algorithm, Control Structures, Flowcharts, Pseudo code, Programming Languages, Generations of Programming Languages.
Database Systems: File Oriented Approach, Database Oriented Approach, Database Views, Three-Schema Architecture, Database Models, Components of DBMS, Introduction of SQL Queries.

Unit – II

Computer Networks: Introduction, Connection Media, Data Transmission Mode, Data Multiplexing, Data Switching, Network Topologies, Types of Networks, Networking Devices, OSI Model.
The Internet: Internet Services, Types of Internet Connections, Internet Security.
Emerging Computer Technologies: Distributed Networking, Peer-to-peer Computing, Grid Computing, Cloud Computing, Utility Computing, On0demand Computing, Wireless Network, Bluetooth, Artificial Intelligence.

Text Reema Thareja, *Fundamentals of Computers*

References

P. K. sinha, *Computer Fundamentals*
Anita Goel, *Computer Fundamentals*
V. Rajaraman, *Fundamentals of Computers*
E. Balagurusamy, *Fundamentals of Computers*
J. Glenn Brookshear, Dennis Brylow, *Computer Science An Overview*

Note: Student friendly video lecturers pertaining to this course are available at **http://spoken-tutorial.org/**
Teachers are advised to teach this courses in the computer lab itself, so that the interested students may derive some time to perform few programs their own.

SEC–4
[A]

Python – 2

BS602

Theory 2 Hours/Week 2 credits

Unit – I

Arrays in Python: Array, Advantages of Arrays, Creating an Array, Importing the Array Module, Indexing and Slicing on Arrays, Processing the Arrays, Types of Arrays, Working with Arrays using numpy, Creating Arrays using array(), linspace, logspace, arange(), zeros() and ones() Functions, Mathematical Operations on Arrays, Comparing Arrays, Aliasing the Arrays, Viewing and Copying Arrays, Slicing and Indexing in numpy Arrays, Dimensions of Arrays, Attributes of an Array, The reshape() Method, The flatten() Method, Working with Multi-dimensional Arrays, Indexing in Multi-dimensional Arrays, Slicing the Multi-dimensional Arrays, Matrices in numpy, Getting Diagonal Elements of a Matrix, Finding Maximum and Minimum Elements, Finding Sum and Average of Elements, Products of Elements, Sorting the Matrix, Transpose of a Matrix, Matrix Addition and Multiplication, Random Numbers.

Strings and Characters: Creating Strings, Length of a String, Indexing in Strings, Slicing the Strings, Repeating the Strings, Concatenation of Strings, Checking Membership, Comparing Strings, Removing Spaces from a String, Finding Sub Strings, Counting Substrings in a String, Strings are Immutable, Replacing a String with another String, Splitting and Joining Strings, Changing Case of a String, Checking Starting and Ending of a String, String Testing Methods, Formatting the Strings, Working with Characters, Sorting Strings, Searching in the Strings, Finding Number of Characters and Words, Inserting Sub String into a String.

Unit – II

Functions: Difference between a Function and a Method, Defining a Function, Calling a Function, Returning Results from a Function, Returning Multiple Values from a Function, Functions are First Class Objects, Pass by Object Reference, Formal and Actual Arguments, Positional Arguments, Keyword Arguments, Default Arguments, Variable Length Arguments, Local and Global Variables, The Global Keyword, Passing a Group of Elements to a Function, Recursive Functions, Anonymous Functions or Lambdas, Function Decorators, Generators, Structured Programming, Creating our Own Modules in Python, The Special Variable name.

Lists and Tuples: List, Creating Lists using range() Function, Updating the Elements of a List, Concatenation of Two Lists, Repetition of Lists, Membership in Lists, Aliasing and Cloning Lists, Methods to Process Lists, Finding Biggest and Smallest Elements in a List, Sorting the List Elements, Number of Occurrences of an Element in the List, Finding Common Elements in Two Lists, Storing Different Types of Data in a List, Nested Lists, Nested Lists as Matrices, List Comprehensions, Tuples, Creating Tuples, Accessing the Tuple Elements, Basic Operations on Tuples, Functions to Process Tuples, Nested Tuples, Inserting Elements in a Tuple, Modifying Elements of a Tuple, Deleting Elements from a Tuple.

Dictionaries: Operations on Dictionaries, Dictionary Methods, Using for Loop with Dictionaries, Sorting the Elements of a Dictionary using Lambdas, Converting Lists into Dictionary, Converting Strings into Dictionary, Passing Dictionaries to Functions, Ordered Dictionaries.

Text R. Nageswara Rao, *Corer Python Programming,* Dreamtech Press

Reference s

Mark Lutz, *Learning Python*
Tony Gaddis, *Starting Out With Python*
Kenneth A. Lambert, *Fundamentals of Python*
James Payne, *Beginning Python using Python 2.6 and Python 3*
Paul Gries,*Practical Programming: An Introduction to Computer Science using Python 3*

Note: Student friendly video lecturers pertaining to this course are available at **http://spoken-tutorial.org/**
Teachers are advised to teach this courses in the computer lab itself, so that the interested students may derive some time to perform few programs their own.

SEC–4 [B]

Information Security

BS602

Theory 2 Hours/Week 2 credits

Unit – I

Introduction to Information Security – Need for Information Security, Threats to Information Systems, Information Assurance, Cyber Security.
Introduction to Application Security and Counter Measures – Introduction to Application Security, Data Security Considerations, Security Technologies, Security Threats, Security Threats to E-Commerce, E-Cash and Electronic Payment System.

Unit – II

Credit/Debit/Smart Cards, Digital Signature, Cryptography and Encryption, Information Security Governance and Risk Management.
Introduction to Security Policies and Cyber Laws – Need for an Information Security Policy, Introduction to Indian Cyber Law, Objective and Scope of the IT Act, 2000, Intellectual Property Issues, Overview of Intellectual-Property- Related Legislation in India, Patent, Copyright.

Text

Dr. Surya Prakash T, Ritendra G, Praveen Kumar S, KLSI, *Introduction to information security and cyber laws (Dreamtech Publication)*

References

Anderson, Ross, *Security Engineering*

G.R.F. Snyder, T. Pardoe, *Network Security*

Mark Stamp, *Information Security: Principles and Practice*

A. Basta, W.Halton, *Computer Security: Concepts, Issues and Implementation*

Mark S. Merkow, Jim Breithaupt, *Information Security: Principles and Practice*

DSC–3F Computer Networks BS605

Theory	3 Hours/Week	3 credits
Practical	2 Hours/Week	1 credit

Unit – I

Introduction: Data Communication Components, Line Configuration, Topologies, Transmission Mode, Categories of Networks, ISO Reference Model–Layered Architecture, Functions of Layers, TCP/IP Reference Model.
Transmission Media: Guided Media–Twisted Pair Cable, Coaxial Cable, Optical Fiber, Unguided Media–Satellite Communication, and Cellular Telephony.
Multiplexing: Frequency–Division Multiplexing, Time–Division Multiplexing.

Unit – II

Data Link Layer: Error Detection–VRC, LRC, CRC, Checksum, Error Correction–Hamming Code, Burst Error Correction, Line Discipline–ENQ/ACK, Poll/Select, Flow Control–Stop-and-Wait, Sliding Window, Error Control–Stop-and-Wait ARQ, Sliding Window ARQ Go-Back-n ARQ, Selective-Reject ARQ.
Local Area Networks: Introduction to IEEE 802, Ethernet-CSMA/CD, Implementation, Token Ring,-Token Passing, Implementation.
Switching: Circuit Switching, Packet Switching, Message Switching.

Unit – III

Networking and Internetworking Devices: Repeaters, Bridges, Routers, Gateways, Brouters, Switches, Distance Vector Routing Algorithm, Link State Routing Algorithm.
Transport Layer: Duties of Transport Layer, Connection.
Upper OSI Layers; Session Layer, Presentation Layer, Application Layer.

Text Behrouz A. Forouzan, *Data Communication and Networking (2e Update)*

References
S.S. Shinde, *Computer Networks*
William Stallings, *Data and Computer Communications*
Andrew S. Tanenbaum, David J Wetherall, *Computer Networks*
Behrouz A Forouzan, Firouz Mosharraf, *Computer Networks A Top-Down Approach*
James F. Kurose, Keith W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*

Networks Lab BS605

Practical 2 Hours/Week 1 credit

1 Write a program to create a socket and implement connect function.

2 Write a program to get MAC address.

3 Write a program to display hello world using signals.

4 Write a program for socket pair system call using IPC.

5 Write a program to implement the sliding window protocol.

6 Write a program to identify the category of IP address for a given IP address.

7 Write a program to print details of DNS host.

8 Write a program to implement listener and talker.

9 Write a program to implement TCP echo using client–server program.

10 Write a program to implement UDP echo using client–server program.

11 Write a UDP client–server program to convert lowercase letters to uppercase letters.

12 Write a TCP client–server program to convert a given string into reverse.

13 Write a UDP client–server program to convert a given string into reverse.

14 Write a program to implement TCP iterative client–server program.

15 Write a program to implement time service using TCP client–server program.

16 Write a program to implement time service using UDP client–server program.

Note : Write above program using C language on Unix/Linux systems.

DSE–1F PHP with MySQL BS606

Theory	3 Hours/Week	3 credits
Practical	2 Hours/Week	1 credit

Unit – I

Introducing PHP – What is PHP? Why use PHP? Evolution of PHP, Installing PHP, Other ways to run PHP, Creating your first script. PHP Language Basics – Using variables, Understanding Data Types, Operators and Expressions, Constants. Decisions and Loops – Making Decisions, Doing Repetitive Tasks with Looping, Mixing Decisions and Looping with HTML.

Strings – Creating and Accessing Strings, Searching Strings, Replacing Text with Strings, Dealing with Upper and Lowercase, Formatting Strings. Arrays – Creating Arrays, Accessing Array Elements, Looping Through Arrays with for-each, Working with Multidimensional Arrays, Manipulating Arrays.

Unit – II

Functions – What is a Function? Why Functions are useful? Calling Functions, Working with Variable Functions, Writing your own Functions, Working with References, Writing Recursive Functions.

Objects – Introduction OOP Concepts, Creating Classes and Objects in PHP, Creating and using Properties, Working with Methods, Object Overloading with _get(), _set() and _call(), Using Inheritance to Extend Power of Objects, Constructors and Destructors, Automatically Loading Class Files, Storing as Strings.

Handling HTML Forms with PHP – How HTML form works, Capturing Form Data with PHP, Dealing with Multi-Value Fields, Generating Web Forms with PHP, Storing PHP Variables in Forms, Creating File Upload Forms, Redirecting After a Form Submission.

Unit – III

Working with Files and Directories - Getting Information on Files, Opening and Closing Files, Reading and Writing to Files, Copying, Renaming, and Deleting Files, Working with Directories.

Introducing Databases and SQL – Deciding How to Store Data, Understanding Relational Databases, Setting Up MySQL, A Quick Play with MySQL, Connecting MySQL from PHP.

Retrieving Data from MySQL with PHP – Setting Up the Book Club Database, Retrieving Data with SELECT, Creating a Member Record Viewer. Manipulating MySQL Data with PHP – Inserting, Updating, and Deleting Records, Building a Member Registration Application.

Text Matt Doyle, *Beginning PHP 5.3* (Wrox – Wiley Publishing)

References

Ellie Quigley, *PHP and MySQL by Example*

Joel Murach, Ray Harris, *Murach's PHP and MySQL*

Brett McLaughlin, *PHP & MySQL: The Missing Manual*

Luke Welling, Laura Thomson, *PHP and MySQL Web Development*

W. Jason Gilmore, *Beginning PHP and MySQL From Novice to Professional*

Andrew Curioso, Ronald Bradford, Patrick Galbraith, *Expert PHP and MySQL*

PHP with MySQL Lab BS606

Practical 2 Hours/Week 1 credit

1 a) Write a PHP script to find the factorial of a given number.
b) Write a PHP script to find the sum of digits of a given number.

2 a) Write a PHP script to find whether the given number is a prime or not.
b) Write a PHP script to demonstrate the use of break, continue statements using nested loops.

3 a) Write a PHP script to display the Fibonacci sequence with HTML page.
b) Write a PHP script to create a chess board.

4 a) Write a PHP script using built-in string function like strstr(), strops(), substr_count(), etc...
b) Write a PHP script to transform a string to uppercase, lowercase letters, make a string's first character uppercase.

5 a) Write a PHP script that inserts a new item in an array in any position.
b) Write a PHP function to check whether all array values are strings or not.

6 a) Write a PHP script to count number of elements in an array and display a range of array elements.
b) Write a PHP script to sort a multi-dimensional array set by a specific key.

7 a) Write a PHP script using a function to display the entered string in reverse.
b) Write a PHP script using function for sorting words in a block of text by length.

8 a) Write a PHP script for creating the Fibonacci sequence with recursive function.
b) Write a PHP script using pass by value and pass by reference mechanisms in passing arguments to functions.

9 a) Write a PHP script to demonstrate the defining and using object properties.
b) Write a PHP script to demonstrate the inheritance.

10 a) Write a PHP script to demonstrate the object overloading with _get(), _set(), and _call().
b) Write a PHP script to demonstrate the overloading property accesses with _get() and _set().

11 a) Write a PHP script to demonstrate the method overloading and method overriding mechanisms.
b) Write a PHP script to demonstrate the use of final classes and final methods.

12 a) Write a PHP script to demonstrate the use interfaces.
b) Write a PHP script using constructors and destructors.

13 Write a PHP application to handling HTML forms with PHP script.

14 a) Write a PHP script to create a file, write data into file and display the file's data.
b) Write a PHP script to check and change file permissions, copying, renaming and deleting files.

15 a) Write a PHP application for connecting to MySQL and reading data from database table.
b) Write a PHP application for inserting, updating, deleting records in the database table.

16 Write a PHP application for student registration form.

DSE–2F Web Technologies BS606

Theory	3 Hours/Week	3 credits
Practical	2 Hours/Week	1 credit

Unit – I

Structuring Documents for the Web: Introducing HTML and XHTML, Basic Text Formatting, Presentational Elements, Phrase Elements, Lists, Editing Text, Core Elements and Attributes, Attribute Groups
Links and Navigation: Basic Links, Creating Links with the <a> Element, Advanced E- mail Links.
Images, Audio, and Video: Adding Images Using the <img> Element, Using Images as Links Image Maps, Choosing the Right Image Format, Adding Flash, Video and Audio to your web pages.
Tables: Introducing Tables, Grouping Section of a Table, Nested Tables, Accessing Tables
Forms: Introducing Forms, Form Controls, Sending Form Data to the Server
Frames: Introducing Frameset, <frame> Element, Creating Links Between Frames, Setting a Default Target Frame Using <base> Element, Nested Framesets, Inline or Floating Frames with <iframe>.

Unit – II

Cascading Style Sheets: Introducing CSS, Where you can Add CSS Rules.
CSS Properties: Controlling Text, Text Formatting, Text Pseudo Classes, Selectors, Lengths, Introducing the Box Model.
More Cascading Style Sheets: Links, Lists, Tables, Outlines, The :focus and :activate Pseudo classes Generated Content, Miscellaneous Properties, Additional Rules, Positioning and Layout with CSS
Page Layout: Understating the Site's Audience, Page Size, Designing Pages, Coding your Design, Developing for Mobile Devices.
Design Issues: Typography, Navigation, Tables, Forms.

Unit – III

Learning JavaScript: How to Add Script to Your Pages, the Document Object Model, Variables, Operators, Functions, Control Statements, Looping, Events, Built- In Objects,
Working with JavaScript: Practical Tips for Writing Scripts, Form Validation, Form Enhancements, JavaScript Libraries.
Putting Your site on the web: Meta tags, Testing your site, Talking the Leap to Live, Telling the World about your site, Understanding your visitors.

Text Jon Duckett, *Beginning HTML, XTML, CSS and JavaScript*

References Chris Bates, *Web Programming*
M. Srinivasan, *Web Technology: Theory and Practice*
Achyut S. Godbole, Atul Kahate, *Web Technologies*
Kogent Learning Solutions Inc, *Web Technologies Black Book*
Ralph Moseley and M. T. Savaliya, *Developing Web Applications*
P.J. Deitel & H.M. Deitel, *Internet and World Wide Web How to program*

Web Technologies Lab BS606

Practical 2 Hours/Week 1 credit

1 a. Write a HTML program using basic text formatting tags, <hn>, <p>, <br>, <pre>.
b. Write a HTML page for Example Cafe using above text formatting tags.

2 a. Write a HTML program using presentational element tags <b>, <i>, <strike>, <sup>, <sub>, <big>, <small>, <hr>
b. Write a HTML program using phrase element tags <blockquote>, <cite>, <abbr>, <acronym>, <kbd>, <address>

3 a. Write a HTML program using different list types.
b. Write a HTML page that displays ingredients and instructions to prepare a recipe.

4 a. Write a HTML program using grouping elements <div> and <span>.
b. Write a HTML Menu page for Example cafe site.

5 a. Write a HTML program using images, audios, videos.
b. Write a HTML program to create your time table.

6 Write a HTML program to create a form using text inputs, password inputs, multiple line text input, buttons, check boxes, radio buttons, select boxes, file select boxes.

7 Write a HTML program to create a frames and links between frames.

8 Write a HTML program to create different types of style sheets.

9 Write a HTML program to create CSS on links, lists, tables and generated content.

10 Write a HTML program to create your college web site using multi column layouts.

11 Write a HTML program to create your college web site using for mobile device.

12 Write a HTML program to create login form and verify username and password using DOM

13 a. Write a JavaScript program to calculate area of rectangle using function.
b. Write a JavaScript program to wish good morning, good afternoon, good evening depending on the current time.

14 a. Write a JavaScript program using switch case?.
b. Write a JavaScript program to print multiplication table of given number using loop.

15 a. Write a JavaScript programs using any 5 events.
b. Write a JavaScript program using JavaScript built in objects.

16 Write a JavaScript program to create registration form and validate all fields using form validation

UG (B.Sc.) Scheme of Examinations
Computer Science
(CBCS 2016-17)

Elaborations

Paper	Credits	Theory Exam		Practical Exam
		University Exam	Internal Exam	
DSC	4	80 Marks	20 Marks	25 Marks
DSE	3	60 Marks	15 Marks	25 Marks
SEC	2	40 Marks	10 Marks	No Practical Exam
GE	2	40 Marks	10 Marks	

DSC – Discipline specific core course
DSE – Discipline specific elective course
SEC – Skill enhancement course
GE – Generic elective

4 Credit Core (DSC) Paper

University Exam (Theory)

Time: 3 Hrs. **Maximum marks: 80**

Section – A (5 X 4M = 20 Marks)

Answer any five of the following eight questions. Each carries four marks.

Q1. From Unit 1
Q2. From Unit 1
Q3. From Unit 2
Q4. From Unit 2
Q5. From Unit 3
Q6. From Unit 3
Q7. From Unit 4
Q8. From Unit 4

Section – B (4 X 15M = 60 Marks)

Answer all the following four questions. Each carries FIFTEEN marks.

Q09. (a) or (b) from Unit 1
Q10. (a) or (b) from Unit 2
Q11. (a) or (b) from Unit 3
Q12. (a) or (b) from Unit 4

Internal Exam (Theory)

Time: 1 Hr. **Maximum marks: 20**

- Two internal exams (one at the middle of the semester and the other at the end) of one-hour duration are to be conducted carrying 15 marks each.
- Average of the scores of two exams should be taken into account.
- Following is the examination pattern.
  - 10 MCQs (multiple choice questions) of half mark each,
  - 10 FIBs (Fill in the Blanks) of half mark each
  - 5 SAQs (short answered questions) of one mark each
  - Totaling 15 marks.
  - 5 marks meant for assignment.

University Exam (Practical)

Time: 2 Hrs. **Maximum marks: 25**

- The question paper is to be typeset with **four** programs with due weightage to **all the units** from the question bank provided in the syllabus.
- The candidates are to answer any **two** of them in the practical exam.
- Each question has to carry **six** marks totaling **12** marks.
- Viva – **8** marks
- Record – **5** marks

3 Credit Core (DSE) Paper

University Exam (Theory)

Time: 3 Hrs. **Maximum marks: 60**

Section – A (5 X 3M = 15 Marks)

Answer any five of the following eight questions. Each carries three marks.

Q1. From Unit 1
Q2. From Unit 1
Q3. From Unit 2
Q4. From Unit 2
Q5. From Unit 3
Q6. From Unit 3
Q7. From Any Unit
Q8. From Any Unit

Section – B (3 X 15M = 45 Marks)

Answer all the following three questions. Each carries fifteen marks.

Q09. (a) or (b) from Unit 1
Q10. (a) or (b) from Unit 2
Q11. (a) or (b) from Unit 3

Internal Exam (Theory)

Time: 1 Hr. **Maximum marks: 20**

- Two internal exams (one at the middle of the semester and the other at the end) of one-hour duration are to be conducted carrying 15 marks each.
- Average of the scores of two exams should be taken into account.
- Following is the examination pattern.
  - 10 MCQs (multiple choice questions) of half mark each,
  - 10 FIBs (Fill in the Blanks) of half mark each
  - 5 SAQs (short answered questions) of one mark each
  - Totaling 15 marks.
  - No assignment is required.

University Exam (Practical)

Time: 2 Hrs. **Maximum marks: 25**

- The question paper is to be typeset with **four** programs with due weightage to **all the units** from the question bank provided in the syllabus.
- The candidates are to answer any **two** of them in the practical exam.
- Each question has to carry **six** marks totaling **12** marks.
- Viva – **8** marks
- Record **– 5** marks

2 Credit Core (SEC & GE) Papers

University Exam (Theory)

Time: 2 Hrs. **Maximum marks: 40**

Section – A (2 X 5M = 10 Marks)

Answer any two of the following four questions. Each carries five marks.

Q1. From Unit 1

Q2. From Unit 1

Q3. From Unit 2

Q4. From Unit 2

Section – B (2 X 15M = 30 Marks)

Answer all the following two questions. Each carries fifteen marks.

Q09. (a) or (b) from Unit 1

Q10. (a) or (b) from Unit 2

Internal Exam (Theory)

Time: 1/2 Hr. **Maximum marks: 10**

- One internal exam at the end of the semester, of half an hour duration is to be conducted carrying 10 marks.
- Following is the examination pattern.
  - 10 MCQs (multiple choice questions) of one mark each,
  - No assignment is required.

MOOCs [Massive Online Open Courses] Free Resources

E-Learning:

NPTEL	:nptel.ac.in	[Core Subjects Certification]
C++ INSTITUTE	:cppinstitute.org	[C++ Certification]
ORACLEEDUCATION	:education.oracle.com	[Java, DBMS Certification]
BIG DATA UNIVERSITY	:bigdatauniversity.com	[Big Data Certification]
COURSERA	:coursera.org	[Core Subjects Certification]
CODEACADEMY	:codecademy.com	[Coding Certification]
KHANACADEMY	:khanacademy.org	[Core Subjects Certification]
PIXAR IN A BOX	:khanacademy.org/partner-content/pixar	
VIDEOLECTURES	:videolectures.net	
YOUTUBEEDU	:plus.google.com/+YouTubeEDU/posts	
DISNEY RESEARCH	:disneyresearch.com	
ALISON	:alison.com	[Core Subjects Certification]
INTERNET ARCHIVE	:archive.org	

Freeware:

SCILAB	: scilab.org	[MatLab Equivalent]
GEOGEBRA	:geogebra.org	[Software for Class Room Teaching]

Search Engine:

WOLFRAM ALPHA	:wolframalpha.com	[Computing Engine]
CITESEER	:citeseerx.ist.psu.edu	[Searching Research Articles]
DOAJ	:doaj.org	[Open Access to Journals]