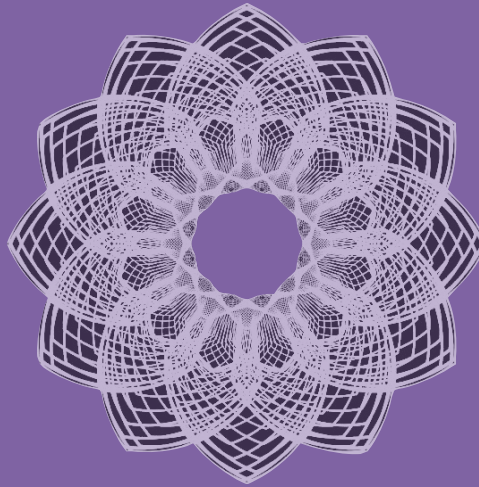# Computer Applications
Syllabus for **B.Sc.**

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

Department of Mathematics
Osmania University
Hyderabad
Telangana

CONTENTS

Syllabus for Computer Applications

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: Python – 1			
BS306	Relational Database Management Systems	DSC–3C	4T+2P=6	4 + 1 =5
	SEMESTER – IV			
BS401	C: SciLab – 2	SEC–2	2T	2
	D: Python – 2			
BS406	Computer Networks	DSC–3D	4T+2P=6	4 + 1 =5
	SEMESTER – V			
BS501	Information Technologies –1	GE–1	2	2
BS502	E: R Basics – 1	SEC–3	2	2
	F: Ruby			
BS505	Multimedia Systems	DSC–3E	3T+2P=5	3 + 1 =4
BS506	Elective–A: Web Technologies	DSE–1E	3T+2P=5	3 + 1 =4
	Elective–B: Visual Programming	DSE–2E		
	SEMESTER – VI			
BS601	Information Technologies –2	GE–2	2T	2
BS602	G: R Basics – 2	SEC–4	2T	2
	H: Ruby on Rails			
BS605	Mobile Applications	DSC–3F	3T+2P=5	3 + 1 =4
BS606	Elective–A: PHP Programming	DSE–1F	3T+2P=5	3 + 1 =4
	Elective–B: Information Security and Cyber Laws	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 and Two Dimensional Arrays, Character Arrays, Functions from ctype.h, string.h.

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, 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), Structures verses Unions, Enumeration Types.

Files: Introduction, Using Files, Working with Text Files and Binary Files, 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 numbers using if and conditional operator.

2   Write a program to calculate arithmetic operations of two numbers using switch.

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

4   Write a program to print whether the given number is a prime or not.

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

6   Write a program to find the sum of two matrices.

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

8   Write a program to print the reverse of a given string.

9   Write a program to find the factorial of a positive integer using iteration and recursion.

10   Write a program to find the GCD of two positive integers using iteration and recursion.

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

12   Write a program to illustrate the use of Enumeration data type.

13   Write a program to illustrate the use of structure concept.

14   Write a program to illustrate the use of union concept.

15   Write a program to write content into a file and display contents of a file

16   Write a program to copy content of one file into another file and display the content of new file.

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.

Unit – II

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

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.

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, 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.

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.

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 print the sum of digits of a given number

2 Write a program to check whether the given number is Armstrong or not

3 Write a program to check whether the given string is Palindrome or not

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

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

6 Write a program to implement inline function inside and outside of a class for
   a. Finding the area of a square
   b. Finding the area of a cube

7 Write a program to implement friend function and friend class

8 Write a program to implement constructor and destructor with in a class.

9 Write a program to demonstrate hierarchical inheritance.

10 Write a program to demonstrate multiple inheritances.

11 Write a program to demonstrate the constructor overloading.

12 Write a program to demonstrate static polymorphism.

13 Write a program to demonstrate dynamic polymorphism.

14 Write a program to implement polymorphism using pure virtual functions.

15 Write a program to demonstrate the function templates and class templates.

16 Write a program to demonstrate exception handling using try, catch, and finally.

Note : 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]

Python – 1

BS301

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

References   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.

DSC–3C   Relational Database Management Systems   BS306

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.

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.

Unit – III

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

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.

Unit – IV

Functional–Dependencies: Anomalies, Partial Functional Dependency, Transitive Functional 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).

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.

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*
	Carlos Coronel, Steven Morris, Peter Rob, *Database Systems: Design, Implementation, and Management*

Relational Database Management Systems Lab   BS306

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

> **Branch** (<u>branchNo</u>, street, city, postcode)
> **Staff** (<u>staffNo</u>, fName, lName, position, sex, DOB, salary, branchNo)
> **PropertyForRent** (<u>propertyNo</u>, street, city, postcode, type, rooms, rent, ownerNo, staffNo, branchNo)
> **Client** (<u>clientNo</u>, fName, lName, telNo, prefType, maxRent, eMail)
> **PrivateOwner** (<u>ownerNo</u>, fName, lName, address, telNo, eMail, password)
> **Viewing** (<u>clientNo</u>, <u>propertyNo</u>, viewDate, comment)
> **Registration** (<u>clientNo</u>, <u>branchNo</u>, 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, lName, 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, if the room is occupied.
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. Investigate the SQL dialect on any DBMS that you are currently using. Determine the system's compliance with the DML statements of the ISO standard. Investigate the functionality of any extensions that the DBMS supports. Are there any functions not supported?
96. Demonstrate that queries written using the UNION operator can be rewritten using the OR operator to produce the same result.
97. Apply the syntax for inserting data into a table.
98. Create a view containing the cheapest hotels in the world.
99. Create the Hotel table using the integrity enhancement features of SQL.
100.   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).

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.

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]

Python – 2

BS401

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
References	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.

DSC–3D   Computer Networks   BS406

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 Concept, CPU Scheduling Concepts, Scheduling Criteria, Overview of Main Memory, Virtual Memory, Mass-Storage Structure, File Systems and File System Implementation.

Unit – II

Introduction: Data Communication Components, Line Configuration, Topologies, Transmission Mode, Categories of Networks, ISO Reference Model–Layered Architecture, TCP/IP Reference Model.

Transmission Media: Guided Media–Twisted Pair Cable, Coaxial Cable, Optical Fiber, Unguided Media–Satellite Communication, and Cellular Telephony.

Unit – III

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. Overview of Multiplexing and Switching.

Unit – IV

Networking and Internetworking Devices: Repeaters, Bridges, Routers, Gateways, Brouters, Switches, Distance Vector Routing Algorithm. Transport Layer: Duties of Transport Layer, Connection.

Upper OSI Layers; Session Layer, Presentation Layer, Application Layer.

Text	Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, *Operating System Concepts (9e)* Behrouz A. Forouzan, *Data Communication and Networking (2e Update)*
References	Andrew S. Tanenbaum, *Modern Operating Systems* Dhananjay M. Dhandhere, *Operating Systems – A Concept Based Approach* 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*

Computer Networks Lab   BS406

Practical   2 Hours/Week   1 credit

1   Use vi editor to create different files, writing data into files, modifying data in files.

2   Use different types of Unix commands on the files created in first program.

3   Write shell programs using 'case', 'then' and 'if' & 'else' statements.

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

5   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.

6   Write a shell script that takes a command–line argument and reports on whether it is directory, a file, or something else.

7   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.

8   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.

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

10   Write a program to display hello world using signals.

11   Write a program to implement the sliding window protocol.

12   Write a program to implement listener and talker.

13   Write a program to implement TCP echo using client–server program

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

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

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

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

Write above program using C language on Unix/Linux Systems.

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]

R Basics– 1

BS502

Theory   2 Hours/Week   2 credits

Unit – I

Introduction – Statistical Programming, The R package, Installation of R and RStudio, Getting started in RStudio

Introduction to the R language – First steps, Basic features of R, Vectors in R, Data storage in R, Packages, libraries, and repositories, Getting help, Logical vectors and relational operators, Data frames and lists, Data input and output.

Unit – II

Programming statistical graphics – High level plots, Low level graphics functions.

Programming with R – Flow control, Managing complexity through functions, The replicate() function, General programming guidelines, Debugging and maintenance.

Text   W. John Braun, Duncan J. Murdoch, *A First Course in Statistical Programming with R (2e)*

References   Jared P. Lander, *R for Everyone*

Joshua F. Wiley, Larry A. Pace, *Beginning R (2e)*

Martin Laredo, *R Programming for Beginners For Data Science*

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]

Ruby

BS502

Theory   2 Hours/Week   2 credits

Unit – I

Ruby Basics – Hello, Matz, Interactive Ruby. Ruby Is Object-Oriented, Ruby's Reserved Words, Variables, Strings, Numbers and Operators.
Conditional Love – The if Statement, The case Statement, The while Loop, The loop Method, The for loop.
Strings – Creating Strings, Concatenating Strings, Accessing Strings, Comparing Strings, Manipulating Strings, Case Conversion, Managing Whitespace, Incrementing Strings, Converting Strings, Regular Expressions.
Math – Class Hierarchy and Included Modules, Converting Numbers, Basic Math Operations, Math Methods, Math Functions.

Unit – II

Arrays – Creating Arrays, Accessing Elements, Concatenation, Set Operations, Comparing Arrays, Changing Elements, Deleting Elements, Multidimensional Arrays.
Hashes – Creating Hashes, Accessing Hashes, Iterating over Hashes, Changing Hashes.
Classes – Defining the Class, Instance Variables, Accessors, Class Variables, Class Methods, Inheritance, Modules, public, private, or protected.

Text   Michael Fitzgerald, *Learning Ruby*

References   Jay McGavren, *Head First Ruby: A Brain-Friendly Guide*
Peter Cooper, *Beginning Ruby From Novice to Professional*
David A. Black, *Ruby Programming (The Well-Grounded Rubyist)*
David Flanagan, Yukihiro Matsumoto, *The Ruby Programming Language*

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.

DSC–3E   Multimedia Systems   BS505

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

Unit – I

Multimedia: Introduction, Definitions, Where to Use Multimedia- Multimedia in Business, Schools, Home, Public Places, Virtual Reality; Delivering Multimedia.

Text: Meaning, Fonts and Faces, Using Text in Multimedia, Computers and Text, Font Editing and Design Tools, Hypermedia and Hypertext.

Images: Before You Start to Create, Making Still Images, Color.

Unit – II

Sound: The Power of Sound, Digital Audio, MIDI Audio, MIDI vs. Digital Audio, Multimedia System Sounds, Audio File Formats. Adding Sound to Your Multimedia Project.

Animation: The Power of Motion, Principles of Animation, Animation by Computer, Making Animations.

Video: Using Video, How Video Works and Is Displayed, Digital Video Containers, Obtaining Video Clips, Shooting and Editing Video.

Unit – III

Making Multimedia: The Stages of a Multimedia Project, the Intangibles, Hardware, Software, Authoring Systems.

The Internet and Multimedia: Internet History, Internetworking, Multimedia on the Web.

Designing for the World Wide Web: Developing for the Web, Text for the Web, Images for the Web, Sound for the Web, Animation for the Web, Video for the Web.

Text   Tay Vaughan, *Multimedia: Making it work (8e)*

References   Keyes, *Multimedia Handbook*

K. Andleigh, K. Thakkar, *Multimedia System Design*

Ralf Steinmetz, Klara Naharstedt, *Multimedia: Computing, Communications Applications*

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

Multimedia Systems Lab   BS505

Practical   2 Hours/Week   1 credit

Implement the followings using Blender -

1 Create an animation using the tools panel and the properties panel to draw the following –
   Line, oval, circle, pencil, brush, lasso tool etc...

2 Create an animation using the tools panel and the properties panel to draw the following –
   rectangle, square, triangle, diamond, octagon etc...

3 Create an animation using text tool to set the font, size, color etc.

4 Create an animation using free transform tool that should use followings-
   Move Objects, Skew Objects, Stretch Objects, Rotate Objects,
   Stretch Objects while maintaining proportion, Rotate Objects after relocating the center dot

5 Create an animation using layers having following features-
   Insert layer, Delete layer, Guide layer, Mask layer.

6 Modify the document (changing background color etc.)using the following tools
   Eraser tool, Hand tool, Ink bottle tool, Zoom tool, Paint Bucket tool, Eyedropper tool

7 Create an animation for bus car race in which both starts from the same point and car wins the race.

8 Create an animation for bus car race in which both starts from the same point and bus wins the race.

9 Create an animation in which text Hello gets converted into GoodBye (using motion/shape tweening).

10 Create an animation in which text gets converted into digits (like hello is 85121215).

11 Create an animation having five images having fade-in fade-out effect.

12 Create an scene to show the sunrise (using multiple layers and motion tweening)

13 Create an scene to show the sunset (using multiple layers and motion tweening)

14 Create an animation to show the ripple effect.

15 Create an animation (using Shape tweening and shape hints) for transforming one shape into another.

16 Create an animation for bouncing ball (you may use motion guide layer).

Note : Practical exercises based on concepts listed in theory using Presentation tools in office automation tool/ GIMP/Blender / Audacity/ Animation Tools/ Image Editors/ Video Editors.

DSE–1E   Web Technologies   BS506

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   BS506

Practical   2 Hours/Week   1 credit

1	a. Write a HTML program using basic text formatting tags, <hn>, <p>,
	, <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

DSE–2E   Visual Programming   BS506

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

Unit – I

Introduction to VB: Writing windows application with VB, Programming languages -procedural, object oriented, event driven; VB Environment, Writing first VB project, compiling, debugging, and running the programs.

Controls : Introduction to controls textboxes, frames, check boxes, option buttons, images, setting borders and styles, the shape control, the line control, working with multiple controls and their properties, designing the user interface, keyboard access, tab controls, default & cancel property, coding for controls.

Variables, constants, and Calculation: Data types, naming rules and conversion, constants-named and intrinsic, declaring variables, scope of variables, val function, arithmetic operations, formatting data Counting and accumulating Sums.

Unit – II

Decisions and Conditions : If statement, Conditions-comparing numeric variables and constants, comparing strings, compound conditions (and, or, not), nested if statements, using if statements with option buttons & check boxes, displaying message in message box, input validation. Calling event procedures, debugging VB projects, Debugging Step-by-Step Tutorial.

Modular programming: Menus, using common dialog box, writing general procedure. Forms Handling : Multiple forms, creating, adding, removing forms, hide, show method, load, unload statement, me keyword, referring to objects on a different forms, Variables and constants in Multiple-Forms.

Iteration Handling: Lists Boxes and Combo Boxes, Do/loops, for/next loops, using msgbox function, using string function.

Unit – III

Arrays: control Arrays, the case structure, single-dimension arrays, for Each/Next statement, table lookup, using list boxes with array, multidimensional arrays.

Database Connectivity: VB and database, using the data control, viewing a database file- step-by-step, Navigating the Database in code, using list boxes and comboboxes as data-bound controls, adding a lookup table and navigation-stepby-step, updating a database file, Recordsets, working with database fields, creating a new Dynaset.

Advanced topics in VB: ActiveX controls, Dynamic link libraries (DLL), Multiple Document interface (MDI).

Text	Julia Case Bradley, Anita C. Millspaugh, *Programming in Visual Basic 6.0 (TMHE 2000–14th Reprint 2004)*
References	Diane Zak, *Programming with Microsoft Visual Basic 2012*
	Tony Gaddis, Kip Irvine, *Starting Out With Visual Basic 2012*

Visual Programming Lab   BS506

1   Write a program to print a table of numbers from 5 to 15 and their squares and cubes.

2   Write a program to print the largest of three numbers.

3   Write a program to print the factional of a number.

4   Write a program to print the GCD of any two positive integers.

5   Write a program to print the given number in reverse order of digits.

6   Write a program to print the given number is prime or not.

7   Create an application that prompts the user to enter today's sales for five stores. The program should then display a simple bar graph comparing each store's sales.

8   Create an application that allows the user to enter each month's amount of rainfall and calculates the total and average rainfall for a year.

9   Write code that declares a string array with three elements and then stores your first, middle, and last names in the array's elements.

10 Enter a list of positive numbers terminated by zero. Find the sum and average of these numbers.

11 A person deposits Rs. 1000 in a fixed account yielding 5% interest.
    Complete the amount in the account at the end of each year for n years.

12 Read n numbers. Count the number of negative numbers, positive numbers and zeros in the list.

13 Read n numbers. Count the number of negative numbers, positive numbers and zeroes in the list. (Use arrays.)

14 Read a single dimension array. Find the sum and average of these numbers.

15 Read a two dimension array. Find the sum of two 2D Array.

16 Create a database Employee and Make a form to allow data entry to Employee Form with the following command buttons:

Employee Form

Employee Nam[   ]

Employee Id: [   ]

Date of Joining[   ]

Designation: [   ]

Department: [   ]

Address: [   ]

Basic Pay: [   ]

[PREV] [NEXT] [FIRST] [LAST] [ADD] [SAVE] [DELETE] [CANCEL]

GE–2   Information Technologies – 2   BS601

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]

R Basics – 2

BS602

Theory   2 Hours/Week   2 credits

Unit – I

Simulation – Generation of pseudorandom numbers, Simulation of other random variables (Bernoulli, Binomial, Poisson, Exponential, Normal), Multivariate random number generation, Markov chain simulation, Monte Carlo integration.

Unit – II

Computational linear algebra – Vectors and matrices in R, Matrix Multiplication, Inversion, Other operations. Numerical optimization – The golden section search method, Newton–Raphson, Built-in functions, Linear programming (Solving LPP in R, Maximization and other kinds of constraints, Special situations).   .

Text   W. John Braun, Duncan J. Murdoch, *A First Course in Statistical Programming with R (2e)*

References   Jared P. Lander, *R for Everyone*

Joshua F. Wiley, Larry A. Pace, *Beginning R (2e)*

Martin Laredo, *R Programming for Beginners For Data Science*

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]

Ruby on Rails

BS602

Theory   2 Hours/Week   2 credits

Unit – I

Getting Started with Rails: What Is Rails? A Brief History of Rails, writing and executing simple rail programs. Understanding Rails: understanding MVC, rails' standard packages, understanding rails' main principles. Ruby's Data Types: Hello, Ruby! Ruby's essential data types, everything is an object, identifiers and variables, working with numbers, Booleans, strings, symbols, regular expressions, ranges, arrays, hashes.

Unit – II

Programming Ruby: defining methods, conditionals – if/elsif/else/unless, ternary operator, case statement, looping – for/in loop, while and until loops, blocks and iterators, exception handling, raising errors, objects and classes, defining and instantiating classes, attributes and accessor methods, methods visibility, single inheritance, monkey patching, singleton methods and eigenclasses. A Working Sample: Creating a New Rails Application, Creating Databases, Scaffolding and Migrations, Putting It All Together: Creating a Rails Application.

Text   Antonio Cangiano, *Ruby on Rails for Microsoft Developers (Bible)*

References   J. Mark Locklear, Learning Rails 5

Noel Rappin, *Professional Ruby on Rails*

Antonio Cangiano*, Ruby on Rails for Microsoft Developers*

David Griffiths, Head First Rails A learner's companion to Ruby on Rails

Michael Hartl, *Ruby on Rails Tutorial Learn Web Development with Rails*

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.

DSC–3F   Mobile Applications   BS605

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

Unit – I

Introduction to Programming and App Inventor: Introduction, What Is a Computer Program? Introducing App Inventor, Getting Hands-On with App, Tutorial 1-1,1-2,1-3,1-4 Working with Media: Displaying Images, Tutorial 2-1,2-2,2-3,Duplicating Blocks and Using Dropdowns, Sounds, Color Blocks, Layout Components, Tutorial 2-7 Input, Variables, and Calculations: The Text Box Component, Performing Calculations, Tutorial 3-1, 3-2 ,Storing Data with Variables Tutorial 3-3, Creating Blocks with Type blocking, Math Functions.

Unit – II

Decision Blocks and Boolean: Introduction to Decision Blocks, Relational Operators and the if Block, Tutorial 4-1, The if then else Block Tutorial 4-2, A First Look At Comparing Strings, Logical Operators, Tutorial 4-4, Nested Decision Blocks, Tutorial 4-5 The if then else if Block, The Screen's Initialize Event, The ListPickerComponent, The CheckBox Component, Repetition Blocks, Times, and Dates: The Notifier Component, The while Loop, Tutorial 5-1, The for each Loop Tutorial 5-2, The Clock Component, The DatePicker Component Procedures and Functions.

Unit – III

Lists -Graphics and Animation: The Canvas Component, Tutorial 9-1, The Ball and ImageSprite Component, Tutorial 9-2, 9-3,Using the Clock Component to Create AnimationsWorking with Text: Concatenating Strings, Comparing Strings, Trimming a String, Converting Case, Finding a Substring Tutorial 10-3,Replacing a Substring , Extracting a Substring, Splitting a Substring Text to Speech and Text Messaging.

Text	Tony Gaddis, Rebecca Halsey, *Starting Out with App Inventor for Android (1e)*

References	Mark L. Murphy, *Beginning Android*
	J.F. DiMarzio, *Android – A Programmer's Guide*
	W Frank Ableson, Robi Sen, Chris King, *Android in Action*
	Lucas Jordan, Pieter Greyling, *Practical Android Projects*
	http://appinventor.mit.edu/

Mobile Applications Lab   BS605

Practical   2 Hours/Week   1 credit

1   Create the Screen for the Hello World App

2   Develop a mobile app to Create Good Morning Translator App

3   Design a mobile app to change the Screen's Background Image

4   Create a mobile app for layout components and Color Blocks

5   Design the mobile app for the Kilometer Converter

6   Create mobile app to calculate Test Average

7   Develop a mobile app to demonstrate Range Checker

8   Develop a mobile app for Grader App

9   Design a mobile app to demonstrate checkbox components

10   Demonstrate a mobile app for while loop

11   Design a mobile app to Calculate Sum of Consecutive Numbers

12   Design a mobile app to create Lights

13   Design a mobile app to demonstrate lists

14   Design a mobile app to validate an Email Address

15   Design a mobile app to display images of all states and union territories in India

16   Design a mobile app of your college having college information, features, events and placements

DSE–1F   PHP Programming   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 –Retrieving Data with SELECT, Creating a Member Record Viewer. Manipulating MySQL Data with PHP – Inserting, Updating, and Deleting Records.

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 Programming   BS606

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   Information Security and Cyber Laws   BS606

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

Unit – I

Introduction to Information Systems and Security – Information Systems, Types of IS, Development of IS, 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.

Unit – II

E-Cash and Electronic Payment System, Credit/Debit/Smart Cards, Digital Signature, Cryptography and Encryption.

Introduction to Security Measures – Secure Information System Development, Application Development Security, Information Security Governance and Risk Management, Security Architecture and Design, Security Issues in Hardware, Data Storage, and Downloadable Devices, Physical Security of IT Assets, Backup Security Measures.

Unit – III

Introduction to Security Policies and Cyber Laws – Need for an Information Security Policy, Information Security Standards – ISO, Introducing Various Security Policies and Their Review Process, 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, Law Related to Semiconductor Layout and Design, Software License.

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*

Information Security and Cyber Laws Lab　　　BS606

Practical　　　　　2 Hours/Week　　　　　1 credit

1　Demonstrate the use of Network tools: ping, ipconfig, ifconfig etc…

2　Demonstrate the use of Network tools: tracert, arp, netstat, whois etc…

3　Use of Password cracking tools: John the Ripper, Ophcrack.
　Verify the strength of passwords using these tools.

4　Write a program for performing encryption and decryption operations of Caesar cipher.

5　Write a program for performing encryption and decryption operations of Rail cipher.

6　Write a program for performing encryption and decryption operations of Monoalphabetic cipher.

7　Write a program for performing encryption and decryption operations of Playfair cipher.

8　Write a program for performing encryption and decryption operations using Transposition technique.

9　Use nmap to analyze a remote machine.

10　Use zenmap to analyze a remote machine.

11　Use Burp proxy to capture and modify the message.

12　Demonstrate sending of a protected word document.

13　Demonstrate sending of a digitally signed document.

14　Demonstrate sending of a protected worksheet.

15　Demonstrate use of steganography tools.

16　Demonstrate use of gpg utility for signing and encrypting purposes.

UG (B.Sc.) Scheme of Examinations
Computer Applications
(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]