CHRIST UNIVERSITY

BANGALORE, INDIA

Declared as Deemed to be University under Section 3 of UGC Act 1956

Department of Computer Science

Proposed Syllabus for

B.Sc. Computer Science

2013

BSc (CME & CMS) – 2013

Semester I

Paper Code	**Paper**	**Hours per Week**	**Marks**	**Credits**
CSC131	Digital Computer Fundamentals and Programming using C	04	100	03
CSC151	C Programming Lab	02	50	01
Total		**06**	**150**	**04**

Semester II

Paper Code	**Paper**	**Hours per Week**	**Marks**	**Credits**
CSC231	Data structures and Operating Systems	04	100	03
CSC251	Data structures Lab	02	50	01
Total		**06**	**150**	**04**

Semester III

Paper Code	**Paper**	**Hours per Week**	**Marks**	**Credits**
CSC331	Data Base Management System and Software Engineering	04	100	03
CSC351	Oracle Lab	02	50	01
Total		**06**	**150**	**04**

Semester IV

Paper Code	**Paper**	**Hours per Week**	**Marks**	**Credits**
CSC431	Object Oriented Programming using JAVA	04	100	03
CSC451	JAVA Programming Lab	02	50	01
Total		**06**	**150**	**04**

Semester V

Paper Code	Paper	Hours per Week	Marks	Credits
CSC531	.Net Technology	03	100	02
Elective I (Choose any one)				
CSC541A	Web Technology	03	100	02
CSC541B	Python Programming			
CSC551	.NET Lab	02	50	01
CSC552A	Web Technology Lab	02	50	01
CSC552B	Python programming Lab			
Total		**10**	**300**	**06**

Semester VI

Paper Code	Paper	Hours per Week	Marks	Credits
CSC631	Computer Networks	03	100	02
Elective I (Choose any one)				
CSC641A	System Software	03	100	02
CSC641B	Computer Architecture			
CSC651	Project Lab	04	100	02
Total		**10**	**300**	**06**

CSC131 Digital Computer Fundamentals and Programming using C

Total teaching Hours/Semester: 60 ***No of Lecture Hours/Week: 04***

I COMPUTER FUNDAMENTALS

Objective

This Subject provides the basic concepts and the functional aspects of a computer. It is starts with the introduction to computers, working principle of a computer and the programming languages. It focuses on different number systems and its applications in computers. The Boolean algebra and different types gate networks, which gives a clear input to the students to understand the basic concepts of flip-flops, combinational logic, registers and counters.

Learning outcome

- To know the working principle of a computer.
- Ability to use Boolean algebra in different number systems and performing computations.
- Simplification of Boolean algebraic expressions.
- Ability to design efficient combinational and sequential logic circuits and implementation.

Unit I. (07)

Introduction to Computers

Characteristics of Computers, Block Diagram of a Digital Computer

Number systems

Different number systems and their conversions (Decimal, Binary, Octal and Hexadecimal) Binary arithmetic - Addition, subtraction, multiplication and division of binary numbers, 1’s and 2’s complement, Floating point numbers, Coding – BCD, Gray, ASCII and EBCDIC.

Unit II. (06)

Boolean Algebra

Boolean operations and expressions, Laws and rules of boolean algebra, Demorgan’s Theorem, Boolean expressions, Simplification of Boolean expression.

Unit III. (06)

Logic Gates

AND gate, OR gate, NOT gate , NAND gate , NOR gate , X-OR gate , X-NOR gate, The universal property of NAND gate and NOR gate,Karnaugh map (SOP).

Unit IV. (05)

Combinational logic

Adders (Half and Full), Decoder, Encoder, Multiplexer , De-Multiplexer (Introductory Concepts Only).

Unit V. **(06)**

Flip-flops

Flip-flops- SR flip-flop, D flip-flop, JK flip-flop, Positive edge triggered flip flops, Master slave JK flip-flop, Introduction to Registers and Counters.

Text Book:

[1] Floyd, Thomas L, *Digital Computer Fundamentals*, 10th Edition, Pearson International, 2009.

Reference Books:

[1] Malvino, Paul Albert, Leach, Donald P. Gautam Saha, *Digital Principles And Applications*, 7th Edition, TMH,2010.

[2] Bartee, Thomas C, *Digital Computer Fundamentals*, 6th Edition, TMH, 2010.

II PROGRAMMING USING C

Objective

To study about algorithms, flowcharts and programs, to solve problems through logical thinking and to learn programming using C.

Learning outcome

- To clearly understand the logic of the problem
- To analyze the given problem and write the algorithm, flowchart
- To write structured C programs

Unit I. **(05)**

Introduction To Programming

Types of Programming, Languages, Structured Programming, Algorithms and Flowcharts with Examples.

Unit II. **(07)**

Introduction To C

History of C- Character set - Structure of a C program – Data types, constants, variables and keywords. Expressions – Statements – Operators – Arithmetic, Unary, Relational and logical, Assignment, Conditional. Library functions. Data Input and output – Single character input, getchar, getch,– Single character output - putchar, Formatted I/O (scanf, printf), gets and puts function.

Unit III. **(08)**

Control Structures And Arrays

Branching: if, if … else , if…else ladder, switch. Looping: while, do…while, for, nested control structures, break, continue statement, goto statement.

Arrays: definition, processing, types - One and Two dimensional arrays. String, string operations (strlen(),strrev(),strcpy(),strcat(),strcmp()strstr()), arrays of strings.

Unit IV. **(05)**

Functions

Functions: Definition, Accessing and prototyping, types of functions, passing arguments to functions, recursion, passing arrays to functions.

Unit V. **(05)**

Pointers And Structure

Pointers: Fundamentals, Declaration, Operations on Pointers, Passing pointers to a function.

Structures : Definition, Processing a structure.

Text Book:

[1] Byron Gottfried, Jitender Chhabra, *Programming with C***,** *3rd* Edition, Tata McGraw-Hill, 2010.

Reference Books:

[1] Balagurusamy E, *Programming in ANSI C,* 4th Edition, Tata-McGraw-Hill, 2007.
[2] Deitel H M and Deitel P J, *C - How to Program*, 7th Edition, Prentice-Hall, 2012.
[3] Yashavant P. Kanetkar, *Let Us C*, 12th Edition, BPB Publications, 2012.
[4].Susant K Rout, *Cimple,* Tata-McGraw-Hill Publishing Company Ltd., 2008.

CSC151 C Programming Lab

Total Hours/Semester: 30 ***No of Hours/Week: 02***

Guidelines:

- Follow Coding standards
- The output of the programs should be neatly formatted
- The programs should be user friendly and interactive
- Use comments wherever necessary

1. Print the size of all the data types with its modifiers supported by C and its range.

2. Convert Fahrenheit to Celsius.

3. Find the roots of a quadratic equation.

4. Accept three numbers and find the largest and second largest among them.

5. Count the number of vowels in a line of text.

6. Accept two numbers and perform various arithmetic operations (+, -, *, /) based on the symbol entered.

7. Print all prime numbers between any 2 given limits.

8. Print all the Armstrong numbers between any 2 given limits.

9. Check whether a string is a Palindrome.

10. Check whether a given matrix is an Identity matrix or not.

11. Perform matrix addition and subtraction.

12. Print first ‘n’ elements of the Fibonacci series.

13. Find the factorial of a number using recursion.

QUESTION PAPER PATTERN

Two questions will be selected by the examiners. Students have to write and execute both the programs.

CSC231 Data structures and Operating Systems

Total teaching Hours/Semester: 60 ***No of Lecture Hours/Week: 04***

I DATA STRUCTURES

Objective

Data Structure is considered as one of the fundamental paper towards a more comprehensive understanding of programming and application development. Student is expected to work towards a sound theoretical understanding of Data Structures.

Learning Outcome

- Understand the need for Data Structures when building application
- Ability to calculate and measure efficiency of code
- Improve programming skills

Unit I. **(05)**

Analysis of Algorithms: Introduction, What to Count and Consider, Rates of Growth, Sequential Search Analysis.

Arrays: Introduction, Array Operations, Merging Arrays, 2D Arrays, Matrix Operations.

Unit II. **(07)**

Linked List: Introduction, Pointers, Insertion, Deletion, Searching, Double Linked List Representations.

Unit III. **(06)**

Stack & Queue: Introduction, Stack Operations using Pointers, Infix to Prefix, Queue Operations using array.

Unit IV. **(05)**

Trees: Introduction, Binary Trees, Properties of Binary Trees, Binary Tree Representations, Binary Tree Traversals.

Unit V. **(07)**

Searching and Sorting: Introduction, Linear Search, Bubble Sort, Binary Search, Insertion Sort, Selection Sort.

Text Book:

[1] Yashwant Kanetkar, *Data Structures Through C,* BPB Publication, 2010.

Reference Books:

[1] Horowitz Sahni Anderson-Freed, *Fundamental of Data Structures in C,* Universities Press, Reprint 2009.

[2] Seymour Lipschultz: *Data Structures,* Schaum series TMH, 2010.

II **OPERATING SYSTEMS**

Objective

To acquire the fundamental knowledge of the operating system architecture and components and to know the various operations performed by the operating system.

Learning outcome

Upon completion of the course students will be able to:

- Understand the basic working process of an operating system.
- Understand the importance of process and scheduling.
- Understand the issues in synchronization and memory management.

Unit I. **(05)**

Introduction and System Structures

Operating system definition, computer system organization, and architecture, structure and operations, process, memory and storage management.

Unit II. **(08)**

Process Management

Process concepts, scheduling and operations on processes. Process Scheduling: Basic concepts, scheduling criteria, scheduling algorithms, Synchronization: Background, critical section problems

Unit III. **(05)**

Deadlock

Deadlock: System model, deadlock characterization, methods for handling deadlock, deadlock prevention, avoidance and detection.

Unit IV. **(06)**

Memory Management

Memory Management Strategies: Background, swapping, Memory allocation, Paging, Structure of the page table.

Unit V. **(06)**

File system

File system: File system structure, directory structure, allocation methods and free-space management. Disk structure, disk scheduling and management.

Text Book:

[1] A. Silberschatz, P.B. Galvin and G. Gagne, *Operating System Concepts*, 8th Edition, New Delhi: Wiley India, 2011.

Reference Books:

[1] Stalling William, *Operating Systems: Internals and Design Principles*, 7th Edition, Prentice Hall, 2011.

[2] Dietel et al, *Operating Systems*, 3rd Edition, Pearson Education, 2004.

[3] A.S. Tanenbaum, *Modern Operating Systems*, 3rd Edition, Prentice Hall, 2007.

CSC251 Data structures Lab

Total Hours/Semester: 30 ***No of Hours/Week: 02***

1. Inserting an element into one dimensional array
2. Deletion of an element in a one dimensional array
3. Implementation of Linear Search
4. Implementation of Binary Search
5. Creation of a linked list and inserting elements into it.
6. Deletion from a linked list.
7. Implementation of different operations on a stack using linked list
8. Implementation of different operations on a queue using an array
9. Implementation of insertion sort
10. Implementation of selection sort.

QUESTION PAPER PATTERN

Two questions will be selected by the examiners. Students have to write and execute both the programs.

CSC331 Database Management System and Software Engineering

Total teaching Hours/Semester: 60 ***No of Lecture Hours/Week: 04***

I DATABASE MANAGEMENT SYSTEM

Objective

To provide strong foundation of database concepts and to introduce students to application development in DBMS.

Learning outcome

- Understanding the fundamentals of RDBMS.
- Building foundation for implementation through project work.

Unit I. (07)

Introduction

Data, Database, Database management system, Characteristics of the database approach, Role of Database administrators, Role of Database Designers, End Users, Advantages of Using a DBMS and When not to use a DBMS.

DBMS Architecture

Data Models – Categories of data models, Schemas, Instances, and Database states. DBMS Architecture and Data Independence – The Three schema architecture, Data independence. DBMS Languages and Interfaces. Classifications of Database Management Systems.

Unit II. (06)

Data Modelling Using Entity-Relationship Model

Using High Level Conceptual Data Models for Database Design, Example Database applications. Entity types, Entity Sets, Attributes and Keys. Relationships, Relationship types, Roles and Structural constraints. Weak Entity Types and Drawing E- R Diagrams

Unit III. (05)

Database Design

Functional dependencies and Normalization for Relational Databases - Normalization concepts, first, second, third normal forms

Unit IV. (07)

SQL

SQL data definition and data types, specifying constraints in SQL, schema change statements, Basic queries, INSERT, DELETE and UPDATE statements in SQL, Views – Concept of a view in SQL.

Unit V. **(05)**

Transaction Processing Concepts and Concurrency Control Techniques

Transaction and System concepts – Desirable properties of Transactions – Schedules and Recoverability. Lock-Based Protocols – Locks, Granting of Locks, and Two phase locking protocol.

Text Book:

[1] Elmasri & Navathe, *Fundamentals of Database Systems,* 5th Edition, Addison – Wesley, 2007.

Reference Books:

[1] O`neil Patric and O`neil Elizabeth, *Database Principles Programming and Performance*, 2nd Edition, Margon Kaufmann Publishers Inc., 2001.

[2] Silberschatz, Korth, Sudarshan, *Database System Concepts*, 5th Edition, McGraw Hill, 2006.

II SOFTWARE ENGINEERING

Objective

To make the students aware of different software engineering principles. To prepare the students to develop the skills necessary to handle software projects. To make the students aware of the importance of software engineering principles in designing software projects.

Learning Outcome

On completion of the course the student shall

- Understand the importance of the stages in the software life cycle.
- Understand the various process models.
- Understand the importance of Software testing.

Unit I. **(05)**
Software and Software Engineering
Nature of software- Defining software, Software Application Domains, Legacy Software-, Software Engineering, The software process, Software Engineering practice – The essence of Practice, General Principles -, Software Myths.

Unit II. **(06)**
Process models
A generic process model – Defining a framework activity, identifying a Task Set, Process Patterns -, Process Assessment and improvement, Prescriptive Process Models – The waterfall Model, Incremental Model, Evolutionary Process Model, Concurrent Models-, A Final Word on Evolutionary Processes.

Unit III. **(06)**
Understanding Requirements
Requirements Engineering, Establishing the groundwork – Identifying Stakeholders, Recognizing multiple viewpoints, Working toward Collaboration, Asking the first questions-, Eliciting requirements – Collaborative requirement gathering, Quality function Deployment, Usage Scenario Elicitation Work Products - , Developing use cases, building the requirements model – Elements of the requirements Model, Analysis pattern -, Negotiating requirements, validating requirements.

Unit IV. **(06)**
Design Concepts
The design within the context of Software Engineering, The design process – Software quality guidelines and attributes, The evolution of software design -, Design concepts – Abstraction, Architecture, Patterns, Separation of concerns, Modularity, information hiding, Functional Independence, refinement, Aspects, Refactoring, Object Oriented design concepts Design classes. The design Model – Data Design elements, Architectural Design elements, Interface Design Elements, Component-Level Design elements, Deployment level Design elements.

Unit V. (07)

Software Testing

A Strategic approach to testing- Verification and Validation, Organizing for software testing, software testing strategy, Criteria for completion of testing-, Test strategies for conventional software – Unit testing, Integration testing- , Test strategies for Object Oriented software-Unit testing in the OO Context, Integration testing in the OO Context-, Validation testing, White-box testing, Basic path testing- Flow Graph Notation, Independent program paths, Deriving test cases, Graph matrices-, control structure testing – Condition testing, Data flow testing, loop testing-, Black-box testing-Graph-based testing methods, Equivalence partitioning, boundary value analysis.

Text Book:

[1] Pressman S Roger, *Software Engineering A Practitioner's Approach*, McGraw Hill International Editions, 7th edition, 2010.

Reference Books:

[1] Sommerville, Ian, *Software Engineering*, Addison Wesley, 9th Edition, 2010.

[2] Rumbaugh, James, *Object Oriented Modeling and design*, Pearson Education, New Delhi, 2005.

CSC351 Oracle Lab

Total Hours/Semester: 30 ***No of Hours/Week: 02***

Objective

The primary objective of this course is to understand relational database concepts and design by using SQL.

Learning outcome

Upon successful completion of the course students will be able to

- Design and implement programming logic for a relational database.
- Manipulate data stored in an Oracle DBMS using Oracle SQL.

1. SQL*Plus and SQL **(04)**
    a. Introduction
    b. Logging on to SQL*Plus and Leaving SQL*Plus
    c. Choosing and Describing Tables
    d. Elements of the SQL Query
    e. Editing SQL Statements
    f. The System Dummy Table
    g. Selecting Columns
    h. Duplicate Information (DISTINCT)
    i. Sorting Information

2. SQL Functions **(04)**
    a. The Concatenation Operator
    b. Column Aliases
    c. String Functions
    d. Arithmetic Functions
    e. Date Functions

3. Advanced SQL Functions **(02)**
    a. Select with Minus, Union and Intersect
    b. Handling NULL

4. Filtering Data Using Where **(02)**
    a. Where Operators
    b. Where with Keywords and Logical Operators

5. Group By and Group By Functions **(02)**
    a. Group Function Examples
    b. Group Function with Having

6. Data Definition Language (DDL) **(04)**

   a. Create, Drop Alter Keywords
   b. Tables
   c. Column
   d. Views
   e. Object
   f. Alter table

7. Data Manipulation Language (DML) **(02)**

   a. Insert
   b. Update
   c. Delete

8. Integrity Constraints **(02)**

   a. Types of Constraint
   b. Referential Integrity
   c. Defining Constraints

9. Retrieving Data from Multiple Tables **(04)**

   a. Joining Tables (Equi-Joins, Non-Equi-Joins)
   b. Aliases for Table Names

10. Sub-Queries **(04)**

   a. Basic Sub queries
   b. Multiple Column Sub queries
   c. Sub queries with Having

CSC431 Object Oriented Programming Using JAVA

Total teaching Hours/Semester: 60 ***No of Lecture Hours/Week: 04***

Objective

The objective of this subject to expose to the students the introduction to OOPs and advantages of object oriented programming. The concepts of OOPs make it easy to represent real world entities.

Learning outcome

- The students are introduced to basics of object oriented programming.
- The students are exposed to have the competence in the use of Java Programming language in the development of small to medium sized application programs that demonstrate professionally acceptable coding.

Unit I. **(12)**

Introduction to OOPs

Problems in Procedure Oriented Approach-Features of Object Oriented Programming System-Class / Object-Encapsulation-Abstraction-Inheritance - Polymorphism. Benefits of OOPs, Applications of OOPs.

Introduction to Java

History-Features of Java-Java and the Internet-How Java differs from C and C++-Java Environment-Structure of Java Program-Java Virtual Machine-Data Types-Constants-Variables-Declaration of variables-Giving values to variables-Scope of variables-Symbolic constants-Literals. Operators-Arithmetic, Boolean logical, Relational and Bitwise operators-Operator Precedence.

Unit II. **(12)**

Classes and String Handling

Classes and Objects - General form of a class-Declaring objects-Accessing class members-Constructors-Parameterized constructors-Overloading constructors-Defining methods-Overloading methods-Returning a value-Recursion-Introducing Access Control-Understanding static-Introducing Final-Garbage collection-finalize() method-this keyword.

Array

Introduction to Arrays, One Dimensional Arrays, Creation of Arrays, Array Intialization, Multidimensional Arrays, arrayname.length, Command Line Arguments.

Unit III. **(12)**

Inheritance And Exception Handling

Basics-Member Access and Inheritance- Super class variable referring to a sub class-Applications of keyword super- Creating a Multilevel Hierarchy-Order of calling constructors-Method Overriding-Dynamic method dispatch-Abstract classes-Using final with Inheritance. Defining an Interface - Implementing interfaces-Variables in interfaces-Extending interfaces.

Exception

Introduction-Types of errors –Exception-Uncaught Exceptions - try and catch - Multiple catch - Nested Try - throw, throws and finally-Built-in Exceptions-User Defined Exception.

Unit IV.

Multithreading And Io Package **(12)**

Multithreaded Programming – Creating Threads-Life cycle of a Thread- Thread Priorities-Synchronization-Inter thread communication-Deadlocks of thread-Avoiding deadlocks in a program –Daemon Threads-Application of Threads.

Io Package

I/O Basics-Streams-Byte Streams-Input Stream classes-Output Stream Classes, Character Streams-Reader Stream classes-Writer Stream classes.

Unit V. **(12)**

Applets

Applet basics-How Applet differs from Applications-Applet Architecture-Applet life cycle-Applet display methods-Repaint-Status window-passing parameters to applets-getDocumentBase() and getCodeBase()-AppletContext and showDocument().

Event Handling-Event handling mechanisms-Delegation Event Model-Event classes-Sources of events-Event listener interfaces-Handling mouse and keyboard events-Adapter classes-Inner classes.

Text Book:

[1] Schildt Herbert, *Java: The Complete Reference*, 8th Edition, Tata McGraw- Hill, 2011.

Reference Books:

[1] E. Balagurusamy, *Programming with JAVA a Primer*, 4th Edition, Tata McGraw-Hill Publishing Company Limited, Delhi, 2010.

[2] Dr.Rao, Nageswara , *Core Java: An Integrated Approach*, Kongent Solutions Inc, 2009.

CSC451 JAVA Programming Lab

Total Hours/Semester: 30 ***No of Hours/Week: 02***

Guidelines

- The output of the programs should be neatly formatted.
- Use meaningful variable names

1. Demonstration the use of methods of Math class.
2. Implementation of various methods of String class
3. Implementation of the concept of array
4. Demonstrate the concept of command line arguments
5. Demonstrate inheritance
6. Implementation of super and this keyword
7. Demonstration of static variables and methods
8. Implementation of interface.
9. Implementation of Abstract class
10. Implementation of Exception Handling
11. Implementation of multithreading by extending Thread class
12. Demonstration of FileInputStream and FileOutput Stream Classes
13. Creation of applets and passing parameters to applets
14. Implementation of Mouse and Keyboard events in an applet.

QUESTION PAPER PATTERN

Two questions will be selected by the examiners. Students have to write and execute both the programs.

CSC531 .Net Technology

Total teaching Hours/Semester: 45 ***No of Lecture Hours/Week: 03***

Objective

To enable the students to learn and develop Web and Windows application for the .NET platform.

Learning outcome

Upon successful completion of the course, the students would be able to

- Update the students with the latest technologies thereby make them fit for the industry
- Make the students aware of a new development platform for internet and distributed applications
- Simplify application development and deployment

Unit I. **(10)**

Introduction

Vision and goals of .NET, Building blocks of .Net, overview of .Net applications, .Net evolution, The .Net Framework Architecture, Intermediate Language(IL), Common Language Runtime (CLR), JIT Compilation, Common Type System (CTS), Common Language System (CLS), Assemblies, IL Disassembler (ILdasm.exe), Namespaces.

C# features

Working with methods- understanding method structure, calling a method, understanding parameter types, overloading methods, virtual methods, overriding methods.

Unit II. **(10)**

C# classes

Constants, fields, methods, properties, events, indexers, operators, constructors, destructors, static modifiers.

Class Inheritance

Compiling with multiple classes, virtual and override methods, abstract methods, sealed classes, Boxing and Unboxing, Working with namespaces, Understanding interfaces, handling exceptions.

Unit III. **(09)**

Windows Applications

Understanding Windows Forms Architecture, Windows controls: Common, Containers, Menus and Tool strips, Data, Reporting. Adding and using windows controls to the form.

Unit IV. **(08)**

Database programming with ADO.NET Understanding the Dataset classes and their relatives, Understanding OLEDB and SQL Server Support, Understanding common database operations using ADO.NET – Operations that don't return rows, Data operations that return single, row entities, data operations that affect single-row entities, data operations returning sets of rows, data operations affecting sets of rows, operations that return hierarchical data.

Unit V. (08)

Creating web applications with web forms [Asp.NET]Difference between ASP and ASP.Net, Defining a web application, ASP.NET architecture, ASP.net web forms, Code behind model, Validation controls in ASP.NET, Server controls and data binding, Grid view, data repeater, data list, Data binding in ASP.NET, Data source controls- sqldata source, Data controls – grid view and details view, Login controls.

Text Book:

[1] Jeff Ferguson, Brian Patterson, Jason Beres ,*C# Programming Bible* , Wiley Publishing Inc., Reprint 2006.

Reference Books:

[1] Jeff Prosise, *Programming .Net,* 2nd Edition, WP Publishers & Distributors Pvt. Ltd, 2009.
[2] Kevin Hoffman & Jeff Gabriel, *Professional .Net Framework*, 1st Edition, Wrox Press Publishers, 2006.

CSC541A Web Technology

Total teaching Hours/Semester: 45 ***No of Lecture Hours/Week: 03***

Objective
Enable students to program for the World Wide Web using HTML, JavaScript, PHP and MySQL.

Learning outcome
Upon successful completion of the course, the students would be able to

- Create static HTML pages and
- Validate HTML FORM data using Javascript at the client side.
- Create dynamic web pages using PHP and MySQL

Unit I. **(08)**
HTML
Introduction to Hyper Text Markup Language, Key components of HTML document, HTML elements, Headers, Linking, Images, Unordered Lists, and Nested and ordered Lists.

Unit II. **(08)**
Tables and Forms
HTML Tables and Formatting, HTML Forms, Internal Linking, Creating and Using Image Maps, frames.

Unit III. **(09)**
JavaScript
Browser and Document object , scripts and HTML Document, variables, expressions, Data type conversions, decisions and loops , control structure, windows Document object, forms and form handling elements, scripting, event handling.

Unit IV. **(10)**
PHP
Essential PHP, Creating a simple PHP program, operators and Flow Control, strings and arrays, creating functions.

Unit V. **(10)**
PHP & MySQL
Connecting to MySQL from PHP, PHP MySQL Connectivity, Creating Databases and Tables with PHP.

Text Books:

[1] Steven M. Schafer, *HTML, XHTML and CSS Bible*, 5th Edition, Wiley-Eastern Publishing Inc., 2011.

[2] Mercer W.Dave, Allan Kent, Steven D Nowichi, David Mercer, Dan Squier, Wankyer Choi *Beginning PHP5*, Wiley – Dreamtech India Pvt. Ltd., 2008.

Reference Book:

[1] Danny Goodman, *JavaScript Bible*, 3rd Edition, Wiley-Eastern Private Ltd., 2010.

CSC541B Python Programming

Total teaching Hours/Semester: 45 ***No of Lecture Hours/Week: 03***

Objective

Learn to program and programming paradigms brought in by Python with a focus on File Handling and Regular Expressions.

Learning Outcomes:

- Able to walk through the algorithm
- Improve programming skills
- Appreciate Python Programming Paradigm
- Hands on Regular Expression
- Write to file handling scripts

Unit I. **(10)**

Introduction and overview

Introduction, What is Python, Origin, Comparison, Comments, Operators, Variables and Assignment, Numbers, Strings, Lists and Tuples, Dictionaries, if Statement, while Loop, for Loop and the range() Built-in Function, Files and the open() Built-in Function, Errors and Exceptions, Functions, Classes, Modules.

Syntax and Style

Statements and Syntax, Variable Assignment, Identifiers, Basic Style Guidelines, Memory Management, Python Application Examples.

Unit II. **(10)**

Python Objects

Python Objects, Standard Types, Other Built-in Types, Internal Types, Standard Type Operators, Standard Type Built-in Functions, Categorizing the Standard Types, Unsupported Types.

Numbers and Strings

Introduction to Numbers, Integers, Floating Point Real Numbers, Complex Numbers, Operators, Built-in Functions. Sequences: Strings, Lists, and Tuples, Sequences, Strings, Strings and Operators, String-only Operators, Built-in Functions, String Built-in Methods, Special Features of Strings.

Unit III. **(08)**

Lists

Operators, Built-in Functions, List Type Built-in Methods, Special Features of Lists, Tuples, Tuple Operators and Built-in Functions, Special Features of Tuples.

Conditionals and Loops

if statement, else Statement, else if Statement, while Statement, for Statement, break Statement, continue Statement, pass Statement, else Statement.

Unit IV. (08)

Files and Input/output

File Objects, File Built-in Function, File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments, File System, File Execution, Persistent Storage Modules

Unit V. (09)

Regular Expressions

Introduction/Motivation, Special Symbols and Characters for REs, REs and Python.

Programming Exercise: Check for data error in CSV files: Numeric Check, Alphanumeric Check, Email Check, Date Check.

Text Book:

[1] Chun, J Wesley, *Core Python Programming, 2nd* Edition, Pearson, 2007 Reprint 2010.

Reference Books:

[1] Barry, Paul, *Head First Python*, 2nd Edition, O Rielly, 2010.
[2] Lutz, Mark, *Learning Python*, 4th Edition, O Rielly, 2009.

CSC551 .NET Lab

Total Hours/Semester: 30 ***No of Hours/Week: 02***

Section A

1. To implement output parameter and reference parameter

2. To implement the concept of indexers

3. To implement the concept of sealed class

4. To implement the concept of namespace

5. To implement the concept of interfaces

6. To implement the concept of events

7. To implement exception handling

Section B

8. To design a calculator in windows form

9. To implement data controls in windows form

10. To implement validation controls in web form

11. To implement Data controls in web form

12. To implement SqlDataReader in ADO.NET

13. To implement Dataset object in ADO.NET

QUESTION PAPER PATTERN

Two questions will be selected by the examiners. Students have to write and execute both the programs.

CSC552A Web Technology Lab

Total Hours/Semester: 30 ***No of Hours/Week: 02***

1. Create a HTML page that will have the following:
   Headers, Linking and Images.
2. Create a HTML page that will have the following:
   Frames, Unordered Lists, Nested and ordered Lists
3. Create a HTML page that will have the following:
   Tables and FORMS
4. FORM validation using Javascript
5. Displaying HTML content using PHP
6. FORM processing using PHP
7. FORM validation using PHP
8. Storing data in MYSQL using PHP
9. Retrieving data from MYSQL using PHP

QUESTION PAPER PATTERN

Two questions will be selected by the examiners. Students have to write and execute both the programs.

CSC552B Python Programming Lab

Total Hours/Semester: 30 ***No of Hours/Week: 02***

1. Create a calculator program.
2. Explore String functions.
3. Implement sequential search.
4. Implement Selection sort.
5. Implement Stack.
6. Creating a CSV File based on user input.
7. Reading a CSV File already created and check for a specific pattern.

QUESTION PAPER PATTERN

Two questions will be selected by the examiners. Students have to write and execute both the programs.

CSC631 Computer Networks

Total teaching Hours/Semester: 45 ***No of Lecture Hours/Week: 04***

Objective

To identify various components in a data communication system and understand state-of-the-art in network protocols, architectures and applications.

Learning outcome

Students will gain

- knowledge about different computer networks
- reference models and the functions of each layer in the models

Unit I. **(07)**

Introduction

Data communications: components – Network criteria – physical structures – network models – categories of networks –interconnection of networks – inter network Protocols and standards: protocols-standards-standards organizations- internet standards Network models: Layered tasks – OSI model – layers in the OSI model – TCP/IP protocol suite.

Unit II. **(10)**

Physical Layer

Data and Signals: Analog and Digital: Analog and Digital Data – Analog and Digital signals, periodic and non periodic signals – Transmission Impairment : Attenuation – Distortion – Noise Data Rate limits : Noiseless channel(Nyquist Bit rate) – Noisy channel (Shannon capacity) Digital Transmission: Digital to digital conversion: Line coding – line coding schemes – block coding - analog to digital conversion – PCM - transmission modes: serial transmission – parallel transmission Analog Transmission: Digital to analog conversion: FSK-ASK-PSK Analog to Analog conversion: Amplitude modulation – Frequency modulation – phase modulation Multiplexing: Frequency division multiplexing – Time division multiplexing – Transmission Media Guided media: Twisted pair cable – coaxial cable – fiber optic cable Unguided media: radio waves - micro waves – infrared.

Unit III. **(10)**

Data link Layer

Error correction and detection: Introduction – block coding – checksum Multiple Access: Random access – Aloha – CSMA – CSMA/CD – CSMA/CA Controlled Access: reservation – polling – token passing Channelization: FDMA - TDMA Wireless LansIEEE 802.11 - architecture – MAC sub layer Addressing mechanism – physical layer - Bluetooth : architecture – Bluetooth layers. Connecting Devices – hubs – repeaters – bridges – switches – routers – gateway.

Unit IV. **(10)**

Network Layer: & Transport Layer

Network Layer: AddressingIPV4 addresses - IPV6 Addresses Internet Protocol:IPv4 – IPv6Address mapping protocols: ARP – RARP Routing protocols: Unicast routing protocols: distance vector routing, Link State routing Multicast Routing protocols (Any two) Transport

Layer: Process to process delivery – UDP – TCP Congestion control and QOS:Data traffic – congestion – congestion control – quality of service – techniques to improve quality of service.

Unit V. (08)

Application layer: & Network Security

DNS: Name space – domain name space – distribution of name space Electronic mail Architecture – FILE transfer: FTP WWW and HTTP: Architecture – web documents – HTTP Network Security: Introduction - definitions – two categories - symmetric keycryptography – traditional ciphers – asymmetric key cryptography.

Text Book:

[1] Behrouz A Forouzan, *Data communication and networking*, 5th Edition, TMH Publications, 2012.

Reference Book:

[1] Andrew S Tanenbaum, Computer Networks, 5th Edition, PHI publication, 2012.

CSC641A System Software

Total teaching Hours/Semester: 45 ***No of Lecture Hours/Week: 04***

Objective

To provide basic knowledge of various system software to get deeper understanding of actual working of a computer system.

Learning Outcome

- Understanding basics of system software.
- Understanding design approaches for developing system software.

Unit I. **(09)**

Introduction

What is System Software? Evolution of Components of a Programming System, General Machine Structure - Memory, Registers, Data and Instructions. Machine Language - No Looping, Address modification using instruction as Data and Index registers, Looping. Assembly Language Program using Literals and pseudo -ops.

Unit II. **(10)**

Assembler

General design procedure, Design of Assembler- Statement of Problem, Data Structures, Format of Databases, Algorithm (2-pass assembler) in brief with flowchart.

Unit III. **(08)**

Macro Language and the Macro Processor

Macro instructions, Features of Macro facility ,Macro instruction argument, Conditional Macro expansions, Macro call within Macro, Implementation.

Unit IV. **(08)**

Loaders and Linker

Loader Schemes- Compile and Go Loader, General Loader scheme, Absolute Loaders, Subroutine Linkages, Relocating Loaders, Direct-Linking Loaders, Binders, Linking loaders, Overlays, Dynamic Binders, Design of an Absolute Loader.

Unit V. **(10)**

Phases of Compilers

Different phases- Lexical Phase, Syntax Phase, Interpretation Phase, Optimization Phase, Storage Assignment Phase, Code Generation Phase and, Assembly phase. Passes of a Compiler with flow chart.

Text Book:

[1] Donovan, John J, *Systems Programming*, Tata McGraw-Hill, Reprint 2001.

Reference Book:

[1] A V Aho and J D Ullman, *Compilers-Principles, Techniques and Tools,* Addison Wesley, 3rd Indian Reprint, 2006.

CSC641B Computer Architecture

Total teaching Hours/Semester: 45 ***No of Lecture Hours/Week: 03***

Objective

The subject deals with basic model of a computer with the concepts on processor design in which data representation and instruction basics are discussed, the basics of design of an ALU, the concepts of memory design.

Learning outcome

At the end of the course students will be able to

- Understand how computer hardware has evolved to meet the needs of multi- processing systems, explain, and demonstrate the concept of top down approach.
- Make a summary on basic operation and state the significant components in computer.
- Explain and demonstrate the basic concept of computer organization and computer architecture.

Unit I. **(07)**

Introduction

Basic Model of a Computer, Computer Components, Register transfer and Micro operations: Register Transfer Language ,Register Transfer , Bus and Memory Transfers, Arithmetic Micro operations , Logic Micro operations , Shift Micro operations , Arithmetic Logic and Shift Unit.

Unit II. **(09)**

Basic Computer organization and design

Instruction codes, Computer registers, Computer Instruction, Timing and control, Instruction cycle, Memory reference instructions, Input output and Interrupt, Design of basic computer, Design of Accumulator logic.

Unit III. **(09)**

Central Processing Unit

Introduction, General Register Organization, Stacks organizations-Register stack, Memory stack, Instruction formats- Three address instruction, two address instruction, one address instruction, zero address instruction , Addressing modes, Data transfer and manipulation- Data transfer instructions, Data manipulation instructions.

Unit IV. **(10)**

Computer Arithmetic

Introduction ,Addition and Subtraction – Addition and subtraction with signed magnitude data, addition and subtraction with signed 2’s complement data ,Multiplication Algorithms-Signed magnitude ,Booth multiplication algorithm, array multiplier ,Division Algorithms- signed magnitude algorithm.

Unit V. (10)
Input Output Organization
Peripheral Device, Input Output Interface – I/O bus and interface modules, I/O versus memory bus, Asynchronous data transfer, Modes of transfer – programmed I/O , Interrupt initiated I/O, , Direct Memory Access – DMA controller and DMA transfer.

Text Book:
[1] Mano M Morris, *Computer System Architecture*, PHI, 3rd Edition, 2008.

Reference Books:
[1] Stalling, Williams. *Computer Organization and Architecture*, 7th Edition, 2010.
[2] Hayes, John P, *Computer Architecture and Organization*, 3rd Edition, McGraw-Hill, 2008.

CSC651 Project Lab

Total Hours/Semester: 60 ***No of Hours/Week: 04***

The main project is to introduce the student to the methodology for solving a problem and preparing a report using the steps of software engineering. Student should take up a project of any domain and submit a report.

Guidelines:

1. The students should carry out the project during the allotted lab hours in the computer lab.
2. The project could be of any domain.
3. The guide can award internal marks by evaluating the performance of the students during the course of the project work.
4. The format of the project report shall be instructed by the faculty in-charge.