School of Applied Sciences

Centurion University of Technology & Management

COURSE STRUCTURE AND SYLLABUS

BACHELOR OF SCIENCE

In

Information Technology

2016

Course Structure

SEMESTER-I				
Sl No	**Subject Code**	**Subject Name**	**Subject Type (L+T+P)**	**Credits**
1	BSFL1101/ FCBS0101	English / Environmental Science	2+0+0	2
2	BSIT1101	PROGRAMMING FUNDAMENTALS USING C/C++	4+0+3	6
3	BSIT1102	COMPUTER SYSTEM ARCHITECTURE	4+0+3	6
4	As per the subject chosen	Generic Elective--1	As per the subject chosen	6
TOTAL CREDITS				20

SEMESTER-II				
Sl No	**Subject Code**	**Subject Name**	**Subject Type (L+T+P)**	**Credits**
1	BSFL1101/ FCBS0101	English / Environmental Science	2+0+0	2
2	BSIT1201	Programming in Java	4+0+3	6
3	BSIT1202	Data Structure	4+0+3	6
4	As per the subject chosen	Generic Elective--2	As per the subject chosen	6
TOTAL CREDITS				20

SEMESTER-III				
Sl No	**Subject Code**	**Subject Name**	**Subject Type (L+T+P)**	**Credits**
1	BSIT2301	Database Management Systems	4+0+3	6
2	BSIT2302	Operating Systems	4+0+3	6
3	BSIT2303	Computer Networks	4+0+3	6
4		Skill Enhancement Course-I	0+0+3	2
5		Generic Elective-3	4+0+3	6
TOTAL CREDITS				**26**

SEMESTER-IV				
Sl no	**Subject Code**	**Subject**	**Contact Hours per week (L+T+P)**	**Credits**
1	BSIT2401	Design and Analysis of Algorithms	4+0+3	6
2	BSIT2402	Software Engineering	4+0+3	6
3	BSIT2403	Internet Technologies	4+0+3	6
4		Skill Enhancement Course-II	0+0+3	2
5		Generic Elective-4	4+0+3	6
TOTAL CREDITS				**26**

Skill Elective Subjects

Sl No	Subject Code	Subject Name	Subject Type (L+T+P)	Credits
1.	BSIT3603	Android Programming	0+0+3	2
2.	BSIT3604	Programming in Python	0+0+3	2
3.	BSIT3605	Hardware and Networking	0+0+3	2
4.	BSIT3606	Realtime Operating System	0+0+3	2
5.	BSIT3607	PHP Programming	0+0+3	2
6.	BSIT3608	Internet Security	0+0+3	2

SEMESTER-V				
Sl No	**Subject Code**	**Subject Name**	**Subject Type (L+T+P)**	**Credits**
1	BSIT3501	Advanced JAVA	4+0+3	6
2	BSIT3502	Theory of Computation	4+0+3	6
3		Discipline Specific Elective-1		6
4		Discipline Specific Elective-2		6
TOTAL CREDITS				**24**

SEMESTER-VI				
Sl No	**Subject Code**	**Subject Name**	**Subject Type (L+T+P)**	**Credits**
1	BSIT3601	Web Services using JAVA	4+0+3	6
2	BSIT3602	Compiler Design	4+0+3	6
3		Discipline Specific Elective-3		6
4		Discipline Specific Elective-4		6
TOTAL CREDITS				**24**

Discipline Specific Elective Subjects

Sl No	Subject Code	Subject Name	Subject Type (L+T+P)	Credits
1.	BSIT3609	Cloud Computing	4+0+3	6
2.	BSIT3610	Big Data Analytics	4+0+3	6
3.	BSIT3611	Network Programming	4+0+3	6
4.	BSIT3612	System Programming	4+0+3	6
5.	BSIT3613	Information Security	4+0+3	6
6.	BSIT3614	Data Mining	4+0+3	6
7.	BSIT3615	Advanced Internet Security	4+0+3	6
8.	BSIT3616	Project	0+0+6	6

Semester- I

BSFL1101 English

Code	Course Title	Credits	L – T – P
BSFL1101	English	2	2 – 0 – 0

The paper in English is of 100 (Hundred) percentage marks.

MODULE-I: Communication Skill **(15 Lectures)**

Communication: Definition, concept
Channels of Communication: Sender, receiver, channel, message, encoding, decoding, context, feedback
Verbal & Non-Verbal Communication: Spoken & written-advantages & disadvantages
Bias free English,
Formal & informal style.

MODULE-II: Communicative Grammar **(16 Lectures)**

Time, Tense & Aspect
Verbs of state & events
Modality
Active & Passive voice
Antonyms, Synonyms, Homonyms, one word substitutions & correction of errors

MODULE-III: Sounds of English **(15 Lectures)**

Length of vowels:
Long vowels as in the words feel, food, shoot, card etc.
Short vowels as in the words pen, sun, cut, shut, etc.
Consonants
Stress pattern
Intonation: Rising & Falling.

Text Books: Effective technical communication by M.A.Rizvi

Reference Books:

Communicative English & Business Communication by R.K.Panda, J.Khuntia, M.Pati, Alok Publication. Communicative Grammar of English Geoffery Leech

FCBS0101 Environmental Science

Code	Course Title	Credits	L – T – P
FCBS0101	Environmental Science	2	2 – 0 – 0

Module-I **(15 Lectures)**

Concepts of Ecology & Environment: Definition-Environment, Ecology & Ecosystem;

Environmental concepts – Atmosphere, Hydrosphere, Lithosphere & Biosphere, Environmental factors – Abiotic factors (Climate & Edaphic) & Biotic factors, Environmental gradients & limiting factor.
Concept of Ecosystem & Processes: Type & Structure, Ecosystem Processes – Energy flow, food chain, food web & ecological pyramids; Biogeochemical cycles – Hydrological cycle(water), gaseous cycle(carbon & oxygen), sedimentary cycle(nitrogen & sulphur).

Module-II **(18 Lectures)**

Population ecology & Ecological succession:
Population ecology: Population density, natality, mortality, population age structure, population growth curves & carrying capacity.
Ecological succession: Characteristics, types (Hydrosere&Xerosere) & Process.
Environmental Pollution: Water pollution, Noise pollution, Air pollution(source, effect, control measure), Depletion of ozone layer – cause, effect & control measure, Green House Effects & Global warming, Acid rain, Biological concentration and biomagnifications, Sewage & sewage treatment.

Module-III **(18 Lectures)**

Conservation of natural resources: Natural resources – renewable, non-renewable, abstract resources, Biodiversity & its conservation, wild life conservation, pollution control board, Environmental awareness & mass education.

Text Books:

Text book of Environmental studies by A.K.Panigrahy & A.Sahu, Sadagrantha Mandir Publishing, Berhampur.

Reference Books:

1. Fundamentals of Ecology by E.P.Odum
2. Environmental Engineering by G.Kiely
3. Fundamentals of Environmental studies by N.K.Tripathy
4. Environmental Biology by P.D.Sharma
5. Ecology & Environment by P.D.Sharma
6. Principles of Environmental Engineering & Science by Davis & Masten

BSIT1101 Programming Fundamentals using C/C++

Code	Course Title	Credits	L – T – P
BSIT1101	Programming Fundamentals using C/ C++	6	4 – 0 – 3

Objective

- Learn problem solving using object-oriented concepts
- Implement object oriented programming using C++
- Realize advantages of object oriented programming over structured programming

Learning outcome

- Able to use object oriented concept to solve problems
- Write an error free program of minimum 200 lines of code

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	Written examination
	Assignment		Report and Presentation
	Experiments	*30*	Practice work, report
	Project		Report and presentation
	Quiz		Surprise/preannounced ones
External Examination		*30*	Written examination
		20	Practice Exam with viva voce
Total		*100*	

Module-I

1. Introduction to C and C++ (3 Lectures)

History of C and C++, Overview of Procedural Programming and Object-Orientation programming, Using main() function, Compiling and Executing Simple Programs in C++.

2. Data Types, Variables, Constants, Operators and Basic I/O (5 Lectures)

Declaring, Defining and Initializing Variables, Scope of Variables, Using Named Constants, Keywords, Data Types, Casting of Data Types, Operators (Arithmetic, Logical and Bitwise),

Using Comments in programs, Character I/O (getc, getchar, putc, putcharetc), Formatted and Console I/O (printf(), scanf(), cin, cout), Using Basic Header Files (stdio.h, iostream.h, conio.hetc).

Module-II

3. Expressions, Conditional Statements and Iterative Statements (5 Lectures)

Simple Expressions in C++ (including Unary Operator Expressions, Binary Operator Expressions), Understanding Operators Precedence in Expressions, Conditional Statements (if construct, switch-case construct), Understanding syntax and utility of Iterative Statements (while, do-while, and for loops), Use of break and continue in Loops, Using Nested Statements (Conditional as well as Iterative)

4. Functions and Arrays (10 Lectures)

Utility of functions, Call by Value, Call by Reference, Functions returning value, Void functions, Inline Functions, Return data type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Command Line Arguments/Parameters in Functions, Functions with variable number of Arguments.

Creating and Using One Dimensional Arrays (Declaring and Defining an Array, Initializing an Array, Accessing individual elements in an Array, Manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Two-dimensional Arrays (Declaring, Defining and Initializing Two Dimensional Array, Working with Rows and Columns), Introduction to Multi-dimensional arrays

5. Derived Data Types (Structures and Unions) (3 Lectures)

Understanding utility of structures and unions, Declaring, initializing and using simple structures and unions, Manipulating individual members of structures and unions, Array of Structures, Individual data members as structures, Passing and returning structures from functions, Structure with union as members, Union with structures as members.

Module-III

6. Pointers and References in C++ (7 Lectures)

Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Pointers to structures, Problems withPointers, Passing pointers as function arguments, Returning a pointer from a function, using arrays as pointers, Passing arrays to functions. Pointers vs. References, Declaring and initializing references, Using references as function arguments and function return values

7. Memory Allocation in C++ (3 Lectures)

Differentiating between static and dynamic memory allocation, use of malloc, calloc and free functions, use of new and delete operators, storage of variables in static and dynamic memory allocation

8. File I/O, Preprocessor Directives (4 Lectures)

Opening and closing a file (use of fstream header file, ifstream, ofstream and fstream classes), Reading and writing Text Files, Using put(), get(), read() and write() functions, Random access in files, Understanding the Preprocessor Directives (#include, #define, #error, #if, #else, #elif, #endif, #ifdef, #ifndef and #undef), Macros

9. Using Classes in C++ (7 Lectures)

Principles of Object-Oriented Programming, Defining & Using Classes, Class Constructors, Constructor Overloading, Function overloading in classes, Class Variables &Functions, Objects as parameters, Specifying the Protected and Private Access, Copy Constructors,Overview of Template classes and their use.

10. Overview of Function Overloading and Operator Overloading (5 Lectures)

Need of Overloading functions and operators, Overloading functions by number and type of arguments, Looking at an operator as a function call, Overloading Operators (including assignment operators, unary operators)

11. Inheritance, Polymorphism and Exception Handling (8 Lectures)

Introduction to Inheritance (Multi-Level Inheritance, Multiple Inheritance), Polymorphism (Virtual Functions, Pure Virtual Functions), Basics Exceptional Handling (using catch and throw, multiple

catch statements), Catching all exceptions, Restricting exceptions, Rethrowing exceptions.

Reference Books

1. HerbtzSchildt, "C++: The Complete Reference", Fourth Edition, McGraw Hill.2003
2. BjarneStroustrup, "The C++ Programming Language", 4th Edition, Addison-Wesley , 2013.
3. BjarneStroustroup, "Programming -- Principles and Practice using C++", 2nd Edition, Addison-Wesley 2014.
4. E Balaguruswamy, "Object Oriented Programming with C++", Tata McGraw-Hill Education, 2008.
5. Paul Deitel, Harvey Deitel, "C++ How to Program", 8th Edition, Prentice Hall, 2011. 5.
John R. Hubbard, "Programming with C++", Schaum's Series, 2nd Edition, 2000.
6. Andrew Koeni, Barbara, E. Moo, "Accelerated C++", Published by Addison-Wesley , 2000. 7.
Scott Meyers, "Effective C++", 3rd Edition, Published by Addison-Wesley, 2005.
8. Harry, H. Chaudhary, "Head First C++ Programming: The Definitive Beginner's Guide", First Create space Inc, O-D Publishing, LLC USA.2014
9. Walter Savitch, "Problem Solving with C++", Pearson Education, 2007.

Programming Fundamentals using C/C++ Lab:

1. WAP to print the sum and product of digits of an integer.
2. WAP to reverse a number.
3. WAP to compute the sum of the first n terms of the following series S
   = 1+1/2+1/3+1/4+……
4. WAP to compute the sum of the first n terms of the following series S
   =1-2+3-4+5…….
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
7. WAP to compute the factors of a given number.
8. Write a macro that swaps two numbers. WAP to use it.
9. WAP to print a triangle of stars as follows (take number of lines from user):

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

10. WAP to perform following actions on an array entered by the user:
    i) Print the even-valued elements
    ii) Print the odd-valued elements
    iii) Calculate and print the sum and average of the elements of array
    iv) Print the maximum and minimum element of array
    v) Remove the duplicates from the array
    vi) Print the array in reverse order
    The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.
11. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
12. Write a program that swaps two numbers using pointers.
13. Write a program in which a function is passed address of two variables and then alter its

contents.

14. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.
15. Write a program to find sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions or new operator.
16. Write a menu driven program to perform following operations on strings:
    a) Show address of each character in string
    b) Concatenate two strings without using strcat function.
    c) Concatenate two strings using strcat function.
    d) Compare two strings
    e) Calculate length of the string (use pointers)
    f) Convert all lowercase characters to uppercase
    g) Convert all uppercase characters to lowercase
    h) Calculate number of vowels
    i) Reverse the string
17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
18. WAP to display Fibonacci series (i)using recursion, (ii) using iteration
19. WAP to calculate Factorial of a number (i)using recursion, (ii) using iteration
20. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.
21. Create Matrix class using templates. Write a menu-driven program to perform following Matrix
    operations (2-D array implementation):
    a) Sum b) Difference c) Product d) Transpose
22. Create the Person class. Create some objects of this class (by taking information from the user).Inherit the class Person to create two classes Teacher and Student class. Maintain the respective information in the classes and create, display and delete objects of these two classes (Use Runtime Polymorphism).
23. Create a class Triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator.
24. Create a class Box containing length, breath and height. Include following methods in it:
    a) Calculate surface Area
    b) Calculate Volume
    c) Increment, Overload ++ operator (both prefix & postfix)
    d) Decrement, Overload -- operator (both prefix & postfix)
    e) Overload operator == (to check equality of two boxes), as a friend function
    f) Overload Assignment operator
    g) Check if it is a Cube or cuboid

    Write a program which takes input from the user for length, breath and height to test the above class.
25. Create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.
26. Write a program to retrieve the student information from file created in previous question and print it in following format:

    Roll No. Name Marks
27. Copy the contents of one text file to another file, after removing all whitespaces.
28. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void

BSIT1102 Computer System Architecture

Code	Course Title	Credits	L – T – P
BSIT1102	Computer System Architecture	6	4 – 0 – 3

Objective

- Grasp characteristics of Computer systems to develop balanced system design that maximizes the processor performance
- Identify the elements of modern instructions sets and their impact on processor design, including how constructs in high-level languages are realized
- Experience use of a design/ simulation tool to model various parts in computer design

Learning outcome

- Capacity to calculate the performance of a modern digital computer from parameters such as processor speed, cycles per instruction
- Understanding of the instruction set of a modern processor
- Formulate instructions of a high-level imperative language in terms of the instruction set of a processor

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	Written examination
	Assignment	*NIL*	Report and Presentation
	Experiments (Internal)	*30*	Practice work, report and viva voce
	Project	*NIL*	Report and presentation
	Quiz	*NIL*	Surprise/preannounced ones
External Examination	Semester	*30*	Written examination
	Experiments (External)	*20*	Practice Exam with viva voce
Total		*100*	

Module-I

1. Introduction **(8 lectures)**

Logic gates, boolean algebra, combinational circuits, circuit simplification, flip-flops and sequential circuits, decoders, multiplexers, registers, counters and memory units.

2. Data Representation and Basic Computer Arithmetic (10 lectures)

Number systems, complements, fixed and floating point representation, character representation, addition, subtraction, magnitude comparison, multiplication and division algorithms for integers

Module-II

3. Basic Computer Organization and Design (13 lectures)

Computer registers, bus system, instruction set, timing and control, instruction cycle, memory reference, input -output and interrupt, Interconnection Structures, Bus Interconnection design of basic computer.

4. Central Processing Unit (15 lectures)

Register organization, arithmetic and logical micro-operations, stack organization, micro programmed control. Instruction formats, addressing modes, instruction codes, machine language, assembly language, input output programming, RISC, CISC architectures, pipelining and parallel architecture.

Module-III

5. Memory Organization (6 lectures)

Cache memory, Associative memory, mapping.

6. Input-Output Organization (8 lectures)

Input / Output: External Devices, I/O Modules, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access, I/O Channels.

Recommended Books:

1. M. Mano, Computer System Architecture, Pearson Education 1992
2. A. J. Dos Reis, Assembly Language and Computer Architecture using C++ and JAVA, Course Technology, 2004 th
3. W. Stallings, Computer Organization and Architecture Designing for Performance, 8 Edition, Prentice Hall of India,2009
4. M.M. Mano , Digital Design, Pearson Education Asia,2013
5. Carl Hamacher, Computer Organization, Fifth edition, McGrawHill, 2012.

Computer System Architecture lab

1. Study of various components of PC.
2. Detail Study of Keyboard and Mouse.
3. Study Anatomy of SMPS
4. Study Anatomy of Motherboard.
5. Some experiments using CPU trainer kits
6. Some experiments using printer trainer kits
7. Dismantling and assembling a PC.
8. Study Anatomy of BIOS
9. Simulation of multiplication algorithm using C / MATLAB
10. Simulation of simple fundamental units like half adder, full adder, multiplexer, de multiplexer,

Arithmetic logic Unit, Simple processor (CPU) etc using VHDL code.

Semester- II

BSFL1101 English

Code	Course Title	Credits	L – T – P
BSFL1101	English	2	2 – 0 – 0

The paper in English is of 100 (Hundred) percentage marks.

MODULE-I: Communication Skill **(13 lectures)**

Communication: Definition, concept
Channels of Communication: Sender, receiver, channel, message, encoding, decoding, context, feedback
Verbal & Non-Verbal Communication: Spoken & written-advantages & disadvantages
Bias free English,
Formal & informal style.

MODULE-II: Communicative Grammar **(15 lectures)**
Time, Tense & Aspect
Verbs of state & events
Modality
Active & Passive voice
Antonyms, Synonyms, Homonyms, one word substitutions & correction of errors

MODULE-III: Sounds of English **(16 lectures)**
Length of vowels:
Long vowels as in the words feel, food, shoot, card etc.
Short vowels as in the words pen, sun, cut, shut, etc.
Consonants
Stress pattern
Intonation: Rising & Falling.

T**ext Books:** Effective technical communication by M.A.Rizvi

Reference Books:
Communicative English & Business Communication by R.K.Panda, J.Khuntia, M.Pati, Alok Publication. Communicative Grammar of English Geoffery Leech

FCBS0101 Environmental Science

Code	Course Title	Credits	L – T – P
FCBS0101	Environmental Science	2	2 – 0 – 0

Course Objectives:

1. To understand the concept of multi-disciplinary nature of Environmental Science where different aspects are dealt with a holistic approach.

2. Students will develop a sense of community responsibility by becoming aware of environmental issues in the larger social context.
3. One must be environmentally educated.

MODULE-I

Environment and its multidisciplinary nature; Need for public awareness; Renewable and non - renewable resources–forest, water, mineral, land, food and energy resources; Structure and function of ecosystems of forest, grass land, desert and aquatic types.

MODULE -II

Biodiversity and its conservation: Biodiversity at global, national and local levels; Threats to biodiversity - Habitat loss; wild life poaching and man - wildlife conflicts; Endangered and endemic species; conservation measures. Causes, effects and control measures of pollution, air, water and noise pollution; Nuclear hazards; solid-waste management–Causes, effects and control measures; Management of disasters due to natural causes of floods, earthquakes, cyclones and landslides.

MODULE-III

Social issues and the environment; Sustainable environment, Water conservation measures; Rain water harvesting; Resettlement and rehabilitation of people; Climate change and global warming; Acid rain; Ozone layer depletion; water land reclamation; Consumerism and waste products; Features of Environment Protection Act, Air pollution and Control of Pollution Acts; Water Pollution and its Control Act. Effects of Pollution explosion on environment and public health; Need for value education to Protect environment and resources.

Learning Outcomes:

1. Understand the natural environment and its relationships with human activities.
2. Characterize and analyze human impacts on the environment.
3. Integrate facts, concepts, and methods from multiple disciplines and apply to environmental problems.
4. Design and evaluate strategies, technologies and methods for sustainable management of environmental systems and for the remediation or restoration of degraded environments.

Text Book:

1. Anubhav Kaushik & C.P. Kaushik: Environmental Studies-New age International Publishers.

Reference Books:

1. Benny Joseph: Environmental Studies-Tata Mac Graw Hill
2. E. Bharucha: Text book of Environmental Studies for under graduate courses– Universities Press. (Book prepared by UGC Committee.

BSIT1201 Programming in Java

Code	Course Title	Credits	L – T – P
BSIT1201	Programming in Java	6	4 – 0 – 3

Objective

- Learn problem solving using object-oriented concepts
- Implement object oriented programming using Java
- Analyze several alternative solutions to determine the best approach

Learning outcome

- Able to use object oriented concept to solve problems
- Write an error free program of minimum 200 lines of code

Evaluation System

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	Written examination
	Assignment	*NIL*	Report and Presentation
	Experiments	*30*	Practice work, report
	Project	*NIL*	Report and presentation
	Quiz	*NIL*	Surprise/preannounced ones
External Examination		*30*	Written examination
		20	Practice Exam with viva voce
Total		*100*	

Module-I

1. Introduction to Java **(4 Lectures)**

Java Architecture and Features, Understanding the semantic and syntax differences between C++ and Java, Compiling and Executing a Java Program, Variables, Constants, Keywords Data Types, Operators (Arithmetic, Logical and Bitwise) and Expressions, Comments, Doing Basic Program Output, Decision Making Constructs (conditional statements and loops) and Nesting, Java Methods (Defining, Scope, Passing and Returning Arguments, Type Conversion and Type and Checking, Built-in Java Class Methods),

2. Arrays, Strings and I/O **(8 Lectures)**

Creating & Using Arrays (One Dimension and Multi-dimensional), Referencing Arrays Dynamically, Java Strings: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability & Equality, Passing Strings To & From Methods, String Buffer Classes. Simple I/O using System.out and the Scanner class, Byte and Character streams, Reading/Writing from console and files.

Module-II

3. Object-Oriented Programming Overview **(4 Lectures)**

Principles of Object-Oriented Programming, Defining & Using Classes, Controlling Access to Class Members, Class Constructors, Method Overloading, Class Variables & Methods, Objects as parameters, final classes, Object class, Garbage Collection.

4. Inheritance, Interfaces, Packages, Enumerations, Autoboxing and Metadata (14 lectures)

Inheritance: (Single Level and Multilevel, Method Overriding, Dynamic Method Dispatch, Abstract Classes), Interfaces and Packages, Extending interfaces and packages, Package and Class Visibility, Using Standard Java Packages (util, lang, io, net), Wrapper Classes, Autoboxing/Unboxing, Enumerations and Metadata.

Module-III

5. Exception Handling, Threading, Networking and Database Connectivity (15 Lectures)

Exception types, uncaught exceptions, throw, built-in exceptions, Creating your own exceptions; Multi-threading: The Thread class and Runnable interface, creating single and multiple threads, Thread prioritization, synchronization and communication, suspending/resuming threads. Using java.net package, Overview of TCP/IP and Datagram programming. Accessing and manipulating databases using JDBC.

6. Applets and Event Handling (15 Lectures)

Java Applets:Introduction to Applets, Writing Java Applets, Working with Graphics, Incorporating Images & Sounds. Event Handling Mechanisms,Listener Interfaces, Adapter and Inner Classes. The design and Implementation of GUIs using the AWT controls, Swing components of Java Foundation Classes such as labels, buttons, textfields, layout managers, menus, events and listeners; Graphic objects for drawing figures such as lines, rectangles, ovals, using different fonts, overview of servlets.

Reference Books

1. Ken Arnold, James Gosling, David Homes, "The Java Programming Language", 4th Edition, 2005.
2. James Gosling, Bill Joy, Guy L Steele Jr, GiladBracha, Alex Buckley"The Java Language Specification, Java SE 8 Edition (Java Series)", Published by Addison Wesley, 2014.
3. Joshua Bloch, "Effective Java" 2nd Edition,Publisher: Addison-Wesley, 2008.
4. Cay S. Horstmann, GaryCornell, "Core Java 2 Volume 1 ,9thEdition,Printice Hall.2012
5. Cay S. Horstmann, Gary Cornell, "Core Java 2 Volume 2 - Advanced Features)", 9th Edition, Printice Hall.2013
6. Bruce Eckel, "Thinking in Java", 3rd Edition, PHI, 2002.
7. E. Balaguruswamy, "Programming with Java", 4th Edition, McGraw Hill.2009.
8. Paul Deitel, Harvey Deitel, "Java: How to Program", 10th Edition, Prentice Hall, 2011. 9.

"Head First Java", Orielly Media Inc. 2nd Edition, 2005.

Programming in Java Lab

1. To find the sum of any number of integers entered as command line arguments
2. To find the factorial of a given number
3. To learn use of single dimensional array by defining the array dynamically.
4. To learn use of .lenth in case of a two dimensional array
5. To convert a decimal to binary number
6. To check if a number is prime or not, by taking the number as input from the keyboard
7. To find the sum of any number of integers interactively, i.e., entering every number from the

keyboard, whereas the total number of integers is given as a command line argument

8. Write a program that show working of different functions of String and StringBufferclasss like setCharAt(, setLength(), append(), insert(), concat()and equals().
9. Write a program to create a ―distance‖ class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer
10. Modify the ―distance‖ class by creating constructor for assigning values (feet and inches) to the distance object. Create another object and assign second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.
11. Write a program to show that during function overloading, if no matching argument is found, then java will apply automatic type conversions(from lower to higher data type)
12. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword
13. Write a program to show the use of static functions and to pass variable length arguments in a function.
14. Write a program to demonstrate the concept of boxing and unboxing.
15. Create a multi-file program where in one file a string message is taken as input from the user and the function to display the message on the screen is given in another file (make use of Scanner package in this program).
16. Write a program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate fibonacii series is given in a different file belonging to the same package.
17. Write a program that creates illustrates different levels of protection in classes/subclasses belonging to same package or different packages
18. Write a program ―DivideByZero‖ that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.
19. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.
20. Write a program to create your own exception types to handle situation specific to your application (Hint: Define a subclass of Exception which itself is a subclass of Throwable).
21. Write a program to demonstrate priorities among multiple threads.
22. Write a program to demonstrate multithread communication by implementing synchronization among threads (Hint: you can implement a simple producer and consumer problem).
23. Write a program to create URL object, create a URLConnection using the openConnection() method and then use it examine the different components of the URLand content.
24. Write a program to implement a simple datagram client and server in which a message that is typed into the server window is sent to the client side where it is displayed.
25. Write a program that creates a Banner and then creates a thread to scrolls the message in the banner from left to right across the applet‘s window.
26. Write a program to get the URL/location of code (i.e. java code) and document(i.e. html file).
27. Write a program to demonstrate different mouse handling events like mouseClicked(), mouseEntered(), mouseExited(), mousePressed, mouseReleased() and mouseDragged().
28. Write a program to demonstrate different keyboard handling events.
29. Write a program to generate a window without an applet window using main() function. 30. Write a program to demonstrate the use of push buttons

BSIT1202 Data Structures

Code	Course Title	Credits	L – T – P
BSIT1202	Data Structures	6	4 – 0 – 3

Objective

- Gain the knowledge of data structure and their implementations
- Understand different algorithms paradigm and method of analysis

Learning outcome

- Develop application using data structure and optimize performance using algorithms
- Analyze algorithms, calculate the complexity and its correctness

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	Written examination
	Assignment	*NIL*	Report and Presentation
	Experiments	*30*	Lab work, report
	Project	*NIL*	Report and presentation
	Quiz	*NIL*	Surprise/preannounced ones
External Examination		*20*	Practice Examination
	Semester	*30*	Semester Examination
Total		*100*	

Module-I

1. Arrays **(5 Lectures)**

Single and Multi-dimensional Arrays, Sparse Matrices (Array and Linked Representation)

2. Stacks **(5 Lectures)**

Implementing single / multiple stack/s in an Array; Prefix, Infix and Postfix expressions, Utility and conversion of these expressions from one to another; Applications of stack; Limitations of Array representation of stack

Module-II

3. Linked Lists **(10 Lectures)**

Singly, Doubly and Circular Lists (Array and Linked representation); Normal and Circular representation of Stack in Lists; Self Organizing Lists; Skip Lists

4. **Queues** **(5 Lectures)**

Array and Linked representation of Queue, De-queue, Priority Queues

5. **Recursion** **(5 lectures)**

Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation)

Module-III

6. **Trees** **(20 Lectures)**

Introduction to Tree as a data structure; Binary Trees (Insertion, Deletion , Recursive and Iterative Traversals on Binary Search Trees); Threaded Binary Trees (Insertion, Deletion, Traversals); Height-Balanced Trees (Various operations on AVL Trees).

7. **Searching and Sorting** **(5 Lectures)**

Linear Search, Binary Search, Comparison of Linear and Binary Search, Selection Sort, Insertion Sort, Insertion Sort, Shell Sort, Comparison of Sorting Techniques

8. **Hashing** **(5 Lectures)**

Introduction to Hashing, Deleting from Hash Table, Efficiency of Rehash Methods, Hash Table Reordering, Resolving collusion by Open Addressing, Coalesced Hashing, Separate Chaining, Dynamic and Extendible Hashing, Choosing a Hash Function, Perfect Hashing Function

Reference Books:

1. Adam Drozdek, "Data Structures and algorithm in C++", Third Edition, Cengage Learning, 2012.
2. SartajSahni, Data Structures, "Algorithms and applications in C++", Second Edition, Universities Press, 2011.
3. Aaron M. Tenenbaum, Moshe J. Augenstein, YedidyahLangsam, "Data Structures Using C and C++:, Second edition, PHI, 2009.
4. Robert L. Kruse, "Data Structures and Program Design in C++", Pearson,1999.
5. D.S Malik, Data Structure using C++,Second edition, Cengage Learning, 2010.
6. Mark Allen Weiss, "Data Structures and Algorithms Analysis in Java", Pearson Education, 3rd edition, 2011
7. Aaron M. Tenenbaum, Moshe J. Augenstein, YedidyahLangsam, "Data Structures Using Java, 2003.
8. Robert Lafore, "Data Structures and Algorithms in Java, 2/E", Pearson/ Macmillan Computer Pub,2003
9. John Hubbard, "Data Structures with JAVA", McGraw Hill Education (India) Private Limited; 2 edition, 2009
10. Goodrich, M. and Tamassia, R. "Data Structures and Algorithms Analysis in Java", 4th Edition, Wiley,2013
11. Herbert Schildt, "Java The Complete Reference (English) 9th Edition Paperback", Tata McGraw Hill, 2014.

Data Structures Lab

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search. Use Template functions.
2. WAP using templates to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.
3. Implement Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists (include a function and also overload operator +).
4. Implement Doubly Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
5. Implement Circular Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
6. Perform Stack operations using Linked List implementation.
7. Perform Stack operations using Array implementation. Use Templates.
8. Perform Queues operations using Circular Array implementation. Use Templates.
9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. WAP to scan a polynomial using linked list and add two polynomial.
11. WAP to calculate factorial and to compute the factors of a given no. (i)using recursion, (ii) using iteration
12. WAP to display fibonacci series (i)using recursion, (ii) using iteration
13. WAP to calculate GCD of 2 number (i) with recursion (ii) without recursion
14. WAP to create a Binary Search Tree and include following operations in tree:
    (a) Insertion (Recursive and Iterative Implementation)
    (b) Deletion by copying
    (c) Deletion by Merging
    (d) Search a no. in BST
    (e) Display its preorder, postorder and inorder traversals Recursively
    (f) Display its preorder, postorder and inorder traversals Iteratively
    (g) Display its level-by-level traversals
    (h) Count the non-leaf nodes and leaf nodes
    (i) Display height of tree
    (j) Create a mirror image of tree
    (k) Check whether two BSTs are equal or not
15. WAP to convert the Sparse Matrix into non-zero form and vice-versa.
16. WAP to reverse the order of the elements in the stack using additional stack.
17. WAP to reverse the order of the elements in the stack using additional Queue.
18. WAP to implement Diagonal Matrix using one-dimensional array.
19. WAP to implement Lower Triangular Matrix using one-dimensional array.
20. WAP to implement Upper Triangular Matrix using one-dimensional array.
21. WAP to implement Symmetric Matrix using one-dimensional array.
22. WAP to create a Threaded Binary Tree as per inorder traversal, and implement operations like finding the successor / predecessor of an element, insert an element, inorder traversal.

Semester- III

BSIT2301 Database Management Systems

Code	Course Title	Credits	L – T – P
BSIT2301	Database Management Systems	6	4 – 0 – 3

Objective

- *To introduce the fundamental concepts of database systems & their importance in practical life and the basic concepts necessary for designing, using and implementing database systems & applications*
- *To make the students understand the principles behind relational database management systems, including the database environment, the relational model, relational languages, develop simple SQL queries*

Learning outcome

- *Demonstrate the underlying concepts of database technology, identify appropriate data model for given problem*
- *Write SQL queries for performing database operations*
- *Design, implement and normalize a relational model for a given problem domain*

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	Written examination
	Experiments	*30*	Lab work, Report and Viva Voce
External Examination	Semester	*30*	Written examination
	Practice (External)	*20*	Practice Exam with viva voce
Total		*100*	

MODULE I: **(16 Lectures)**

Introduction to database Systems, Basic concepts & Definitions, Data Dictionary, DBA, File-oriented

system vs. Database System, Database Language.
Database System Architecture-Schemas, Sub Schemas & Instances, 3-level database architecture, Data Abstraction, Data Independence, Components & functions of DBMS, Data models, Mapping E-R model to Relational, Network and Object Oriented Data models.

MODULE-II: (15 Lectures)

Relational Algebra, Tuple & Domain Relational Calculus, Relational Query Languages: SQL and QBE.
Database Design:-Database development life cycle (DDLC), Functional dependency and Decomposition, Dependency Preservation & lossless Design, Normalization, Normal forms: 1NF, 2NF, 3NF, and BCNF, Multi-valued Dependencies, 4NF & 5NF.
Query processing and optimization: Evaluation of Relational Algebra Expressions.

MODULE -III: (18 Lectures)

Transaction processing and concurrency control: Transaction concepts, concurrency control, locking and Timestamp methods for concurrency control.
Database Recovery System: Types of Data Base failure & Types of Database Recovery techniques. Advanced topics: Object-Oriented & Object-Relational Database, Parallel & Distributed Database, Data warehousing & Data Mining.

Text Books:

1. Database Systems by Thomas Connolly and Carolyn Begg-Pearson Education-3rd Edition
2. Fundamentals of Database System by Elmasari &Navathe- Pearson Education-5th

References Books:

(1) An introduction to Database System – Bipin Desai, Galgotia Publications
(2) Database System Concepts by Sudarshan, Korth (McGraw-Hill Education)

DATABASE MANAGEMENT SYSTEMS LAB (0-0-3)

1. Introduction to Database and Database languages.
2. Use of SQL syntax: insertion, deletion using SQL.
3. Use of SQL syntax: updation, modification using SQL.
4. Programs on join statements and SQL queries including where clause.
5. Programs on procedures and functions.
6. Programs on database triggers.
7. Programs on packages.
8. Programs on data recovery using check point technique.
9. Concurrency control problem using lock operations.
10. Programs on JDBC and ODBC using database.

Text Books:
1. PL/SQL by Ivan Bayross - BPB Publication

BSIT2302 OPERATING SYSTEMS

Code	Course Title	Credits	L – T – P
BSIT2302	Operating Systems	6	4 – 0 – 3

Objective

- To understand the various process communication, synchronization and scheduling techniques
- To understand the various memory management and virtual memory management techniques
- To have kernel level coding practice

Learning outcome

- Customize Linux Kernel
- Implement modified scheduling algorithms

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	Written examination
	Assignment	-	Report and Presentation
	Experiments	*30*	Lab work, report
	Project	-	Report and presentation
	Quiz	-	Surprise/preannounced ones
External Examination		*50*	Written examination
Total		*100*	

MODULE-I **(15 Lectures)**

INTRODUCTION TO OPERATING SYSTEM:
What is an Operating System? Simple Batch Systems, Multiprogramming and Time Sharing systems Parallel Systems, Distributed Systems and Real time Systems.
Operating System Structures: Operating System Services, System components, Protection system, Operating System Services, system calls.

PROCESS MANAGEMENT:
Process Concept, Process Scheduling, Operation on Processes, Inter-process communication, Examples of IPC Systems, Multithreading Models, Threading Issues, Process Scheduling Basic concepts, scheduling criteria, scheduling algorithms, Thread Scheduling.

MODULE-II **(17 Lectures)**

PROCESS COORDINATION: Synchronization: The Critical section problem, Peterson's solution, Synchronization hardware, Semaphores, Classical problems of synchronization.
Deadlocks: System model, Deadlock Characterization Methods for Handling Deadlocks, Deadlock Prevention, Deadlock avoidance, Deadlock Detection, recovery from Deadlock.
MEMORY MANAGEMENT: Memory Management strategies, Logical versus Physical Address space, swapping, Paging, Segmentation.
Virtual Memory: Background, Demand paging, performance of Demand paging, Page Replacement, Page replacement algorithms. Allocation of frames, Thrashing, Demand Segmentation.

MODULE-III **(17 Lectures)**

STORAGE MANAGEMENT:
File System Concept, Access Methods, File System Structure, File System Structure, File System Implementation, Directory implementation, Efficiency and Performance, Recovery, Overview of Mass Storage Structure, Disk Structure, Disk Scheduling, Disk Management, Swap-Space Management.
CASE STUDIES: The LINUX System, Windows XP, UNIX system.

TEXT BOOK:
1. Operating System Concepts – Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, 8th edition, Wiley-India, 2009.
2. Modern Operating Systems – Andrew S. Tanenbaum, 3rd Edition, PHI
3. Operating Systems: A Spiral Approach – Elmasri, Carrick, Levine, TMH Edition

REFERENCE BOOK:
1. Operating Systems – Flynn, McHoes, Cengage Learning
2. Operating Systems – Pabitra Pal Choudhury, PHI
3. Operating Systems – William Stallings, Prentice Hall
4.Operating Systems, - Rohit Khurana, 1st Edition, Vikas Publishing House Pvt Ltd.

OPERATING SYSTEM LAB THROUGH UNIX/LINUX (0-0-3)

1. Detail anatomy of Operating System.
2. Basic DOS Commands and its Use.
3. Basic UNIX / LINUX commands and its Use.
4. Study of different editors in LINUX (vi, gedit, etc.)
5. Detail study of File Access Permission in LINUX.
6. Detail study of UNIX Shell Programming.
7. Programs on process creation and synchronization, inter process communication including shared memory, pipes and messages. (Dinning Philosopher problem / Cigarette Smoker problem / Sleeping barber problem).
8. Programs on UNIX System calls.

9. Simulation of CPU Scheduling Algorithms. (FCFS, RR, SJF, Priority, Multilevel Queuing).
10. Simulation of Banker's Algorithm for Deadlock Avoidance, Prevention.
11. Program for FIFO, LRU, and OPTIMAL page replacement algorithm.

BSIT2303 Computer Networks

Code	Course Title	Credits	L – T – P
BSIT2303	Computer Networks	6	4 – 0 – 3

Objective

- *Understand the basic principles of computer networks*
- *Practice the programming aspects of computer networks*

Learning outcome

- *Develop chat application using socket programming*
- *Client-Server programming*

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	One Written examination One Practice examination Assignment/Presentation
	Assignment		Report and Presentation
	Experiments	*30*	Lab work, report
	Project		Report and presentation
	Quiz		Surprise/preannounced ones
External Examination		*(30+20)*	Written examination
Total		*100*	

MODULE-I **(18 Lectures)**

Overview of Computer Networks:
Introduction: OSI, TCP/IP and other networks models, Examples of Networks: Novell Networks,Arpanet, Internet, Network Topologies WAN, LAN, MAN.
Physical Layer: Transmission mode, Transmission Media: Guided Media, Unguided media (wireless), Multiplexing: FDM, WDM, TDM, Circuit switching and Telephone Network: Circuit switching, Telephone network.

MODULE-II **(16 Lectures)**

Data link layer: Design issues, framing, error detection and correction, CRC, Elementary Protocol-stop and wait, Sliding Window, Data link layer in HDLC,ATM.
Point-to –Point Access: PPP, Multiple Access Protocols: Random Access, Controlled Access, Channelization.
Local area Network: Ethernet.
Wireless LANs: IEEE 802.11, Bluetooth virtual circuits

MODULE-III **(18 Lectures)**

Network Layer : addressing and Network Layer Protocols: ARP, IPV4, ICMP, IPV6 ad ICMPV6, Broad cast, Multi cast, Congestion, Control Algorithms – General Principles of Congestion prevention policies. Internetworking: The Network layer in the internet and in the ATM Networks.
Transport Layer: Process to Process Delivery: UDP; TCP congestion control.
Application Layer:
Client Server Model, Domain Name System (DNS): Electronic Mail (SMTP) and file transfer (FTP) HTTP and WWW.

Text Books:
1. Data Communications and Networking: Behrouz A. Forouzan, Tata McGraw-Hill, 4thEd
3. Computer Networks: A. S. Tannenbum, D. Wetherall, Prentice Hall, Imprint of Pearson 5thEd

Reference Book : .
1. Computer Networks:A system Approach:Larry L, Peterson and Bruce S. Davie,Elsevier, 4thEd
2. Computer Networks: Natalia Olifer, Victor Olifer, Willey India
3. An Engineering Approach to Computer Networks-S.Keshav, 2nd Edition, Pearson Education
4. Computer Networking: A Top-Down Approach Featuring the Internet, James F. Kurose and Keith W. Ross , 2nd Edition, Pearson Education, 2002.

COMPUTER NETWORKS LAB (0-0-3)

Some Network protocol simulation using NetSim, NS2, etc. for

1. Analysing bus vs. star-switch with respect to number of collisions (for a fixed number of transmitting nodes) for Ethernet LAN
2) Analysing number of transmitting nodes vs. collision count, mean delay for Ethernet LAN
3) Analysing performance of token ring with number of nodes vs. response time, mean delay using NetSim.
4) Comparing the throughput and normalized throughput for token ring and token bus for different transmitting nodes.
5) Comparing the CSMA/CD vs. CSMA/CA protocols (for a fixed number of transmitting nodes).
6) Analysing the difference between unicast and broadcast transmission (for a fixed number of transmitting nodes).
7) Verification of stop-and-wait protocol.
8) Verification of Go-back-N protocol.
9) Verification of Selective repeat protocol.
10) Verification of distance vector and link state routing algorithm.

Semester- IV

BSIT2401 Design and Analysis of Algorithm

Code	Course Title	Credits	L – T – P
BSIT2401	Design and Analysis of Algorithm	6	4 – 0 – 3

Objective

- Gain the knowledge of data structure and their implementations
- Understand different algorithms paradigm and method of analysis

Learning outcome

- Develop application using data structure and optimize performance using algorithms
- Analyze algorithms, calculate the complexity and its correctness

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	Written examination
	Assignment	*NIL*	Report and Presentation
	Experiments	*30*	Lab work, report
	Project	*NIL*	Report and presentation
	Quiz	*NIL*	Surprise/preannounced ones
External Examination		*20*	Practice Examination
	Semester	*30*	Semester Examination
Total		*100*	

MODULE-I **(16 Lectures)**

Introduction to analysis and design of algorithm, Growth of functions, Asymptotic notations, Recurrences, Solution of recurrences by substitution, Recurrence tree and the master method. Divide and conquer algorithms (Worst case analysis of merge sort, quick sort and heap sort algorithms), Priority queue, Data structure for disjoint sets (Disjoint set operations, linked list representation, disjoint set forests)

MODULE-II **(16 Lectures)**

Dynamic programming approach: Matrix chain multiplication, longest common subsequence.
Greedy method: Fractional knapsac problem Greedy verses dynamic programming, Huffman codes. Concept of backtracking, branch & bound design techniques. Graph algorithms: Minimal spanning tree (Kruskal and Prim's algorithms), Single source shortest paths (Bellman-Ford and Dijkstra's algorithm), Floyd's algorithm.

MODULE –III **(16 Lectures)**

Flow Network, Ford-Fulkerson method, Fast Fourier Transform, Rabin-Karp string matching algorithm. NP-Completeness, Polynomial time solvability, Verification and Reducibility, NP complete problems (without proof), Approximation algorithm for the traveling salesman problem.

Text book:

1. T.H. Cormen, C.E. Leiserson, R.L. Rivest and L.Stein, "Introduction to Algorithms" , Second Edition, PHI Learning, 2002
Chapters: 1, 2, 3, 4(excluding 4.4), 6, 7 (7.4.1), 15(15.2, 15.3. 15.4), 16(16.1, 16.2, 16.3),

21(21.1, 21.2, 21.3) 23, 24(24.1, 24.2, 24.3), 26(26.1, 26.2), 30(30.1, 30.2), 32(32.1, 32.2), 34, 35(35.2)

Reference books:

1. E. Horowitz, S. Sahani, S. Rajsekharan, "Fundamentals of Computer Algorithms", Second Edition, Universities Press, 2007
2. J. Kleinbers, E.Tardos, Algorithm design, Pearson Education Inc., New Delhi , 2006
3. R. Johnsonbaugh, M. Schaefer, "Algorithms", Pearson Education Inc., New Delhi , 2004
4. Kenneth A. Berman & Jerome L. Paul, "Algorithms", Revised Edition, 2005, CENGAGE Learning India Pvt. Ltd., New Delhi.
5. Anany V. Levitin, "Introduction to the Design and Analysis of Algorithms", Second Edition, 2007, Pearson Education Inc., New Delhi.
6. Michael T. Goodrich and Roberto Tamassia, "Algorithm Design: Foundations, Analysis, and Internet Examples", 2nd Edition, Wiley India Pvt. Ltd., New Delhi

Design and Analysis of Algorithms Lab (0-0-3)

1. i. Implement Insertion Sort (The program should report the number of comparisons)
   ii. Implement Merge Sort(The program should report the number of comparisons)
2. Implement Heap Sort(The program should report the number of comparisons)
3. Implement Randomized Quick sort (The program should report the number of comparisons)
4. Implement Radix Sort
5. Create a Red-Black Tree and perform following operations on it:
   i. Insert a node
   ii. Delete a node
   iii. Search for a number & also report the color of the node containing this number.
6. Write a program to determine the LCS of two given sequences
7. Implement Breadth-First Search in a graph
8. Implement Depth-First Search in a graph
9. Write a program to determine the minimum spanning tree of a graph

For the algorithms at S.No 1 to 3 test run the algorithm on 100 different inputs of sizes varying from 30 to 1000. Count the number of comparisons and draw the graph. Compare it with a graph of nlogn.

BSIT2402 Software Engineering

Code	Course Title	Credits	L – T – P
BSIT2402	Software Engineering	6	4 – 0 – 3

Objective

- Learn software development processed: user requirements, specifications, design, coding, testing, maintenance, documentation, management
- Learn DevOps software development model

Learning outcome

- Identify, collect requirements and prepare design documents required for managing a software
- Build software using DevOps model so as to release the software faster and more frequently

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	Written examination
	Assignment	*NIL*	Report and Presentation
	Experiments	*30*	Lab work, report
	Project	*NIL*	Report and presentation
	Quiz	*NIL*	Surprise/preannounced ones
External Examination		*20*	Practice Examination
	Semester	*30*	Semester Examination
Total		*100*	

MODULE-I **(16 Lectures)**

The Evolving Role of Software, Software Characteristics, Changing Nature of Software, Software Engineering as a Layered Technology, Software Process Framework, Framework and Umbrella Activities, Process Models, Capability Maturity Model Integration (CMMI).

Software Requirement Analysis, Initiating Requirement Engineering Process, Requirement Analysis
and Modeling Techniques, Flow Oriented Modeling, Need for SRS, Characteristics and Components of SRS.

MODULE-II **(18 Lectures)**

Estimation in Project Planning Process, Project Scheduling.
Software Risks, Risk Identification, Risk Projection and Risk Refinement,RMMM Plan.
Quality Concepts, Software Quality Assurance, Software Reviews,Metrics for Process and Projects.

MODULE-III **(16 Lectures)**

Design Concepts, Architectural Design Elements, Software Architecture,Data Design at the Architectural Level and Component Level, Mapping of Data Flow into Software Architecture, Modeling Component Level Design.
Software Testing Fundamentals, Strategic Approach to SoftwareTesting, Test Strategies for Conventional Software, Validation Testing, System testing, Black-Box Testing, White-Box Testing and their type, Basis Path Testing.

Recommended Books:

1. R.S. Pressman, Software Engineering: A Practitioner's Approach (7th Edition), McGraw-Hill, 2009.
2. P. Jalote, An Integrated Approach to Software Engineering (2nd Edition), Narosa Publishing House, 2003.
3. K.K. Aggarwal and Y. Singh, Software Engineering (2nd Edition), New Age International Publishers, 2008.
4. I. Sommerville, Software Engineering (8th edition), Addison Wesley, 2006.
5. D. Bell, Software Engineering for Students (4th Edition), Addison-Wesley, 2005.
6. R. Mall, Fundamentals of Software Engineering (2nd Edition), Prentice-Hall of India,2004

SOFTWARE ENGINEERING LAB (0-0-3)

S. No	Practical Title
1.	• Problem Statement, • Process Model
2.	Requirement Analysis: • Creating a Data Flow • Data Dictionary, Use Cases
3.	Project Management: • Computing FP • Effort • Schedule, Risk Table, Timeline chart
4.	Design Engineering: • Architectural Design • Data Design, Component Level Design
5.	Testing: • Basis Path Testing

Sample Projects

1. **Criminal Record Management**: Implement a criminal record management system for jailers, police officers and CBI officers
2. **DTC Route Information**: Online information about the bus routes and their frequency and fares
3. **Car Pooling**: To maintain a web based intranet application that enables the corporate employees within an organization to avail the facility of carpooling effectively.
4. Patient Appointment and Prescription Management System
5. Organized Retail Shopping Management Software
6. Online Hotel Reservation Service System
7. Examination and Result computation system
8. Automatic Internal Assessment System
9. Parking Allocation System
10. Wholesale Management System

BSIT2403 Internet Technologies

Code	Course Title	Credits	L – T – P
BSIT2403	Internet Technologies	6	4 – 0 – 3

Objective

- Familiar with client server architecture and able to use the skills for web project development

Learning outcome

- Develop a static, interactive and well-formed webpage using Javascript, CSS3 and HTML5

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	Written examination
	Assignment	*NIL*	Report and Presentation
	Experiments	*30*	Lab work, report
	Project	*NIL*	Report and presentation
	Quiz	*NIL*	Surprise/preannounced ones
External Examination		*20*	Practice Examination
	Semester	*30*	Semester Examination
Total		*100*	

MODULE-I **(16 Lectures)**

Java
Use of Objects, Array and Array List class
JavaScript
Data types, operators, functions, control structures, events and event handling.

MODULE-II **(15 Lectures)**

JDBC
JDBC Fundamentals, Establishing Connectivity and working with connection interface, Working with statements, Creating and Executing SQL Statements, Working with Result Set Objects.

MODULE-III **(18Lectures)**

JSP

Introduction to JavaServer Pages, HTTP and Servlet Basics, The Problem with Servlets, The Anatomy of a JSP Page, JSP Processing, JSP Application Design with MVC, Setting Up the JSP Environment, Implicit JSP Objects, Conditional Processing, Displaying Values, Using an expression to Set an Attribute, Declaring Variables and Methods, Error Handling and Debugging, Sharing Data Between JSP Pages, Requests, and Users, Database Access.

Java Beans

Java Beans Fundamentals, JAR files, Introspection, Developing a simple Bean, Connecting to DB

Recommended Books:

1. Ivan Bayross, Web Enabled Commercial Application Development Using Html, Dhtml,javascript, Perl Cgi , BPB Publications, 2009.
2. Cay Horstmann, BIG Java, Wiley Publication , 3rd Edition., 2009
3. Herbert Schildt , Java 7, The Complete Reference, , 8th Edition, 2009.
4. Jim Keogh ,The Complete Reference J2EE, TMH, , 2002.
5. O'Reilly , Java Server Pages, Hans Bergsten, Third Edition, 2003.

Create event driven program for following:

1. Print a table of numbers from 5 to 15 and their squares and cubes using alert.
2. Print the largest of three numbers.
3. Find the factorial of a number n.
4. Enter a list of positive numbers terminated by Zero. Find the sum and average of these numbers.
5. A person deposits Rs 1000 in a fixed account yielding 5% interest. Compute the amount in the account at the end of each year for n years.

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

Skill Elective-1

Nomenclature

Subject Name	**Code**	**Type of course**	**(L+T+P)**	**Prerequisite**
Android Programming	BSIT3603	Practice	0+0+3	Programming in Java

Objective

- Learn to develop Android mobile Apps

Learning outcome

- Develop Android mobile Apps using Kotlin

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*NIL*	Written examination
	Assignment	*NIL*	Report and Presentation
	Experiments (Internal)	*50*	Practice work, report and viva voce

	Project	*NIL*	Report and presentation
	Quiz	*NIL*	Surprise/preannounced ones
External Examination	Semester	*NIL*	Written examination
	Experiments (External)	*50*	Practice Exam with viva voce
Total		*100*	

Module – I **(16 hrs.)**

Introduction to Android OS, Development environment (Android Studio, Kotlin), Android application basics (Activity, Service, Content provider, Broadcast receiver, Intents)

Module – II **(24 hrs.)**

User Interface – Layouts, Widgets, Event handling, Notifications

Data Storage –Shared preference, Internal storage, External storage, SQLite, Content provider

Module – III **(20 hrs.)**

Networking – Android network overview and management, Socket and HTTP, Wi-Fi and Bluetooth, GPS; Handler, AsyncTask

Multimedia – Voice recording, Image capturing, Drawing & animation

Reference

Text Books:

1. Antonio Leiva, *Kotlin for Android Developers*. Leanpub, 2018.
2. Ed Burnette, *Hello, Android: Introducing Google's Mobile Development Platform*. O′Reilly, 2010.

Reference Books:

1. Pradeep Kothari, *Android Application Development, Black Book*. Kogent Learning Solutions Inc, 2014.
2. G. Blake Meike, *Programming Android*. O′Reilly, 2011.
3. Michael Burton, *Android App Development for Dummies*. John Wiley & Sons, 2015.

Online Sources:

1. https://developers.google.com/training/android/
2. https://developer.android.com/kotlin/
3. https://kotlinlang.org/docs/tutorials/kotlin-android.html
4. http://www.vogella.com/tutorials/android.html
5. https://www.tutorialspoint.com/android/

Skill Elective-2

Nomenclature

Subject Name	**Code**	**Type of course**	**(L+T+P)**	**Prerequisite**
Programming in Python	BSIT3604	Practice	0+0+3	Nil

Objective

~~**Key points:**~~

- Learn problem solving using object-oriented concepts
- Implement object oriented programming using Python

Learning outcome

- Able to use object oriented concept to solve problems
- To quickly and easily draw plot or visualize the information through visualization Technique
- Write an error free program of minimum 200 lines of code

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*NIL*	Written examination
	Assignment	*NIL*	Report and Presentation
	Experiments (Internal)	*50*	Practice work, report and viva voce
	Project	*NIL*	Report and presentation
	Quiz	*NIL*	Surprise/preannounced ones
External Examination	Semester	*NIL*	Written examination
	Experiments (External)	*50*	Practice Exam with viva voce
Total		*100*	

Module I **(16 Hrs)**

Familiarization of python: Features and Installation, Setting up Path, Working with Python Basic Syntax, Variable and Data types, Basic operators, Numbers, Array, Tuples, Dictionary, Date and Time.

String Manipulation: Accessing Strings, Basic Operations, Stringslices.

Introduction of List: Accessing lists, Operations, Working with lists List (Functions).

Decision Making Statement: Boolean Expressions, if-else Statement.

Iteration Statement: The while Statement, for Statement, Nested Loops, Break and Continue statement.

Functions Operation: Types of Function, Recursion.

Object Oriented Concept: OOPS Terminologies, Defining Classes, Creating Objects, Regular expressions, Constructors, Inheritance and Overloading.

- **Files Operation:** Create Text & Binary Files

Module II **(17 Hrs)**

GUI Programming: Example GUI Program, Environment Variables, Label, Message Widget, Text Area, Button, Radio Button, CheckBox, Listbox/DropDown Box, Frames, Menu Widget, Menu Button Widget, Scrollbar, Forms: GET method and POST method, Sliders (Tkinter), Uploading files, Database access, Sending email.

Formatted: Normal, Don't add space between paragraphs of the same style, Line spacing: Multiple 1.15 li, No bullets or numbering, Widow/Orphan control, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted Table

Formatted: Space Before: 0 pt, Don't add space between paragraphs of the same style, Line spacing: Multiple 1.15 li, Outline numbered + Level: 1 + Numbering Style: Bullet + Aligned at: 0.63 cm + Indent at: 1.27 cm, Widow/Orphan control, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted Table

Formatted: List Bullet, Line spacing: 1.5 lines

Formatted: List Bullet, Line spacing: 1.5 lines

Formatted: Font: (Default) Times New Roman, 11 pt

Formatted: Indent: Left: 0 cm, Line spacing: 1.5 lines

Formatted: Line spacing: 1.5 lines, No bullets or

Formatted: No Spacing, Line spacing: 1.5 lines

- **Module III** **(16 ~~h~~Hrs)**

- **Data Visualization**: Visualization Libraries, Data frame: Data types, Attributes, methods (mean(),median(),std(),var(),cor(),min()/max(),describe()),groupby method, Selecting Column, Filtering, Selecting row and column, Missing values, Data read from excel, CSV and txt file.

Plotting: Basic plot(), Histogram, Bar Plot, Box Plots, Area Plot, Scatter Plot, Pie Chart.

-

Skill Elective-3

Nomenclature

Subject Name	**Code**	**Type of course**	**(L+T+P)**	**Prerequisite**
Hardware and Networking	BSIT3605	Practice	0+0+3	Nil

Objective

- Develop ability to understand the anatomy of Computer and peripherals
- To have an overall idea about Networking concepts and Components
- To have an overall idea about Secure Computing

Learning outcome

- Build your own Device
- Troubleshoot various faults in a computer system or network
- Construct small LAN for resource sharing

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	-	Written examination
	Assignment	-	Report and Presentation
	Experiments	*50*	Assessment based on Individual Lab sessions, Lab report
	Project	-	Report and presentation
	Quiz	-	Surprise/preannounced ones
External Examination	*Practice*	*50*	Lab Test.
Total		*100*	

Course outline **(45 ~~h~~Hrs)**

I. **Inside The PC:** Opening a Desktop/Laptop and Identification of Various Components, Study of Different Blocks of PC.

II. **Assembling and Disassembling of PC:** Assembling and Disassembling of Different Components like Motherboard, Processor, Ram, Hard Disc and Other Components.

III. **SMPS and Peripherals:** Identification of Various Power Supply Units and Peripherals along with Pin

Formatted: Line spacing: 1.5 lines, No bullets or

Formatted: Indent: Left: 0 cm, Line spacing: 1.5 lines

Formatted: Line spacing: 1.5 lines

Structure.

IV. **BIOS:** Study of Basic Input Output System, Morden BIOS (UEFI), Security Features.

V. **Installation of OS and Application Software:** Installation of Ubuntu LTS (Latest), Application Software.

VI. **Usage of Libreoffice:** Writer (Document), Calc (Spreadsheet), Impress (Presentation).

VII. **Basic Commands**: Working With Linux Commands, Vi Editor, Shell Scripting.

VIII. **Basic Networking:** Wired LAN, Wireless LAN, Wireless Security and Troubleshooting.

IX. **Introduction to Various Networking Devices:** Routers, Switches, Modems, Hubs, Firewall and wireless devices.

X. **Network Configuration:** Setting IP Addresses, Sharing Files and Folders, Network Troubleshooting, PING Test, Ipconfig etc.

XI. **Introduction To Servers and Network Security:** Files Servers, Email Servers, Proxy Servers; Essential Security Measures: Encrypt Critical Data, Secure Systems With Passwords, Back Up and Isolate Information.

Skill Elective-4

Nomenclature

Subject Name	**Code**	**Type of course**	**(L+T+P)**	**Prerequisite**
Realtime Operating System	BSIT3606	Practice	0+0+3	Nil

Objective

- Student will be able to understand and design real time operating systems

Learning outcome

- Can develop embedded applications in Real time environment.

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	-	Written examination
	Assignment	-	Report and Presentation
	Experiments	*50*	Assessment based on Individual Lab sessions, Lab report
	Project	-	Report and presentation
	Quiz	-	Surprise/preannounced ones
External Examination	*Practice*	*50*	Lab Test.
Total		*100*	

Course outline **(45 ~~h~~Hrs)**

I. Design and Real Time Implementation of Real time Video Surviellance System
II. Real time Implementation of Stepper Motor controller
III. Design and Implementation to operate calculator using Touch screen
IV. Design and Implementation of Edge Detection in real time
V. Design and Implementation of Touch Screen Controller on FPGA using Nanoboard
VI. Design and Implementation of a PWM Controller and control the Speed of the Motor
VII. Design a Real Time clock.
VIII. Interface a Seven Segment Display
IX. Design and Implementation of Keyboard controller on the Nanoboard
X. SMS control Block using Nanoboard
XI. Implement a Ethernet controller and test it on to the board

Skill Elective-5

Nomenclature

Subject Name	**Code**	**Type of course**	**(L+T+P)**	**Prerequisite**
PHP Programming	BSIT3607	Practice	0+0+3	Web Technology

Objective

- On completion of this course, a student will be familiar with Content Management System (WordPress).

Learning outcome

- Develop an Interactive, Dynamic website/ web application

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*NIL*	Written examination
	Assignment	*NIL*	Report and Presentation
	Experiments	*50*	Lab work, report
	Project	*NIL*	Report and presentation
	Quiz	*NIL*	Surprise/preannounced ones
External Examination		*50*	Practice Examination
Total		*100*	

Module I **(15 Hrs)**

PHP7: PHP7, Environment Setup, Declarations, Operator, Return Type, Decision and Looping, Arrays, Anonymous Classes, Functions, Browser Control and Detection, String, Form Processing, Files, Cookies, Sessions and Exception Handling.

Module II **(18 Hrs)**

JSON : Intro, syntax, JSON with Ajax

AJAX - Technologies, Browser Support, Action, XMLHttpRequest, Database Operations

PHP MySql: Connection with MySQL, Command, Execute, Fetch, Insert, delete, and update records. Sending Email, File Upload, OOPs in PHP, PHP & Ajax, introduction Cake PHP.

Module III **(18 Hrs)**

Content Management System (CMS) – WordPress (WordPress 4.9.6 or Higher Version): Installation, Configuration, Template Creation, Uploading a Template, Select a Template, Install and Activate a Theme,Plan Website By Choosing Color Schemes, Fonts, Layouts, Add Images, Add Photo Galleries, Use Plugin, Customize a WordPress site using PHP7

Skill Elective-6

Nomenclature

Subject Name	**Code**	**Type of course**	**(L+T+P)**	**Prerequisite**
Internet Security	BSIT3608	Practice	0+0+3	Programming in Java

Objective

- In Depth study of various network attacks techniques and methods to defend against them.

Learning outcome

- Expertise in different network attacks techniques and methods to defend.

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*NIL*	Written examination
	Assignment	*NIL*	Report and Presentation
	Experiments	*50*	Lab work, report
	Project	*NIL*	Report and presentation
	Quiz	*NIL*	Surprise/preannounced ones
External Examination		*50*	Practice Examination
Total		*100*	

- **Course outline** **(45 ~~h~~Hrs)**

Formatted: Indent: Left: 0 cm, Hanging: 0.63 cm, Line spacing: 1.5 lines, No bullets or numbering

1. **Introduction and Overview**
   - Internet Architecture
   - How the Internet works (high-level overview)
   - IP Address
2. **TCP/IP Protocols, Vulnerabilities, Attacks, and Countermeasures**
   - Physical Layer: jamming attacks
   - Data Link Layer: ARP protocol and ARP cache poisoning
   - Network Layer: IP protocols, packet sniffering, IP Spoofing, IP fragmentation attacks

- Network Layer: ICMP protocol and ICMP misbehaviors
- Network Layer: IP Routing protocols and Attacks
- Transport Layer: TCP protocol, TCP session hijacking, reset and SYN flooding attacks
- DoS and DDoS attacks
- DNS protocol and attacks
- BGP protocol and Attacks

3. **Network Security Mechanisms**
   - IP Tunneling and SSH Tunneling
   - Virtual Private Networks
   - Firewalls
   - Bypassing firewalls
   - Transport Layer Security (TLS/SSL)
   - TLS Programming

Semester- V

BSIT3501 Advanced Java

Code	*Course Type*	*Credits*	*L-T-P*
BSIT3501	***Theory + Practice***	***6***	***4-0-3***

Objective

- Familiar with JDBC architecture

Learning outcome

- Develop a dynamic, interactive and well-formed enterprise application.

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	Written examination
	Assignment	*NIL*	Report and Presentation
	Experiments	*30*	Lab work, report
	Project	*NIL*	Report and presentation
	Quiz	*NIL*	Surprise/preannounced ones
External Examination		*20*	Practice Examination
	Semester	*30*	Semester Examination
Total		*100*	

MODULE-I

(25 Lectures)

Introduction to JDBC, JDBC architecture, java.sql Package, Connection, Statement, ResultSet, Prepared Statement, Callable Statement, Scrollable and Updatable ResultSet, Batch Updates, ResultSetMetaData, Simple Transaction Management, Four Levels of JDBC drivers, their pros & cons, Features of JDBC 3.0/4.0

MODULE-II

(30 Lectures)

Need of Server side Programming, Introduction to Servlets, Servlet Life Cycle, javax.servlet package, ServletConfig, ServletContext, ServletResponse, Supplying initialization parameters to Servlets, Performing database operations in Servlets, Include and forward mechanisms, Applying filters to Servlets, javax.servlet.http Package, HttpServlet Life Cycle, Http request methods GET vs POST, HttpServletRequest, HttpServletResponse, Dealing with Http headers & error codes, Session Tracking, purpose, Hidden form fields, Cookies, Http Session, URL rewriting, Event listeners, Web application security.

MODULE-III

(25 Lectures)

Disadvantages of Servlets, Introduction to JSP, JSP Life Cycle, Creating dynamic Web content with JSP Scripting elements, Scriptlet, Declaration, Expression, XML syntax for JSP elements, JSP directives page, include and taglib, JSP implicit objects, JSP scopes, Include and forward mechanism, Using a Java bean in a JSP, JSP Model 1 architecture, JSP Model 2 (MVC) architecture, Custom Tag Development, Classic Tags, Simple Tags, Error Handling in a JSP, JSTL, Expression Language, Processing XML in a JSP.

TEXT BOOKS:

1. *Advanced Java Programming by Uttam K Roy (Author), Oxford Publication*

REFERENCE BOOKS:

1. *Advanced Java 2 Platform: How to Program by Deitel publication.*
2. *JDBC Database Access with Java: A Tutorial and Annotated Reference Book by Graham Hamilton and Roderic Geoffrey Galton Cattell.*
3. *Java servlet and JSP cookbook Book by Bruce W. Perry.*

BSIT3502 Theory of Computation

Code	*Course Type*	*Credits*	*L-T-P*
BSIT3502	***Theory + Practice***	***6***	***4-0-3***

Objective

- Learn several formal mathematical models of computation along with their relationships with formal languages.
- Learn regular languages and context free languages which are crucial to understand how compilers and programming languages are built.
- Strengthen their rigorous mathematical reasoning skills.

Learning outcome

After studying this course, Students will get knowledge on:

- Design and development principles in the construction of software systems of varying complexity.
- Demonstrate knowledge of basic mathematical models of computation and describe how they relate to formal languages.

Evaluation Systems

Internal Examination	*Component*	*% of Marks*	*Method of Assessment*
	Midterm Test	*20*	Written examination
	Assignment	*NIL*	Report and Presentation
	Experiments	*30*	Lab work, report
	Project	*NIL*	Report and presentation
	Quiz	*NIL*	Surprise/preannounced ones
External Examination		*20*	Practice Examination
	Semester	*30*	Semester Examination
Total		*100*	

Module I: **(28 Hours)**

Introduction: Automata,Alphabet, languages and grammars. Production rules and derivation oflanguages. Chomsky hierarchy of languages.
Finite automata: Finite automata (deterministic and nondeterministic). Minimization of finite automata. Equivalence between two FA's.
Regular grammars: Regular grammars, regular expressions. Closure and decision properties of regular sets.Pumping lemma of regular sets.
Context free grammar: Grammar**,** Parse trees and derivations,Left and right linear grammars.

Module II: **(26 Hours)**

More about Context free grammar: Normal forms: Chomsky Normal Form (CNF), Greibach Normal Form (GNF). Cook, Younger, Kasami (CYK) and Early's parsing algorithms. Ambiguity and properties of context free languages. Pumping lemma, closure properties of context free languages.
Pushdown automata: Introduction to Pushdown automata. Types (deterministic and nondeterministic).
Linear bounded automata: Introduction, Linear bounded automata and context sensitive languages.

Module III: **(23 Hours)**

Turing machine: Turing machines and variation of Turing machine model, Turing computability. Turingcomputable functions. Church Turing hypothesis.
Recursive Languages: Recursive and recursively enumerable sets. Decidable languages and undecidableproblems, Halting problem.
Universal Turing machine: Universal Turing machine. Turing Reducibility. Valid and invalid computationsof Turing machines.
Practice(using JFLAP)

1. *Conversion of Non-deterministic finite automaton (NFA) to Deterministic finite automaton (DFA).*
2. *Conversion of FA to Regular Expression.*
3. *Converting FA to Grammar.*
4. *Conversion of Pushdown Automata to Context free Grammar.*
5. *Implementation of Turing machine.*

Text Books:

1. *Introduction to Automata Theory, Languages and Computation: J.E. Hopcroft and J.D Ullman, Pearson Education,3rd Edition.*
2. *Introduction to Computer Theory, by Daniel I.A. Cohen, john wiley and sons inc.*

Semester- VI

BSCIT3601 Web services using JAVA

Code	*Course Type*	*Credits*	*L-T-P*
BSCIT3601	***Theory + Practice***	***6***	***4-0-3***

Objective

- Understand the design rationale for SOAP based web services in terms of objectives, trade-offs and alternatives.

Learning outcome

- Design and implement Web Services that accommodates specified functional and non-functional requirements and constraints.

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	Written examination
	Assignment	*NIL*	Report and Presentation
	Experiments	*30*	Lab work, report
	Project	*NIL*	Report and presentation
	Quiz	*NIL*	Surprise/preannounced ones
External Examination		*20*	Practice Examination
	Semester	*30*	Semester Examination
Total		*100*	

MODULE-I
Java Web Services Introduction **(25 Lectures)**

What Are Web Services? What Good Are Web Services? A First Example, The Service Endpoint Interface and Service Implementation Bean, A Java Application to Publish the Web Service, Testing the Web Service with a Browser, A Perl and a Ruby Requester of the Web Service, The Hidden SOAP,A Java Requester of the Web Service Messages, What's, Key Features of the First Code Example, Java's SOAP API, An Example with Richer Data Types, Publishing the Service and Writing a Client, Multithreading the Endpoint Publisher.

All About WSDLs
What Good Is a WSDL? Generating Client-Support Code from a WSDL, The @Web Result Annotation, WSDL Structure, A Closer Look at WSDL Bindings, Key Features of Document-Style Services, Validating a SOAP Message against a WSDL's XML Schema, The Wrapped and Unwrapped Document Styles, Amazon's E-Commerce Web Service, An E-Commerce Client in Wrapped Style. An E-Commerce Client in Unwrapped Style, Tradeoffs Between the RPC and Document Styles, An Asynchronous E-Commerce Client, thewsgen Utility and JAX-B Artifacts, A JAX-B Example, Marshaling and wsgen Artifacts, An Overview of Java Types and XML Schema Types, Generating a WSDL with the wsgen Utility, WSDL Wrap-Up, Code First Versus Contract First, A Contract-First Example with wsimport, A Code-First, Contract-Aware Approach, Limitations of the WSDL.

MODULE-II
SOAP Handling **(30 Lectures)**

SOAP: Hidden or Not? SOAP 1.1 and SOAP 1.2,SOAP Messaging Architecture, Programming in the JWS Handler Framework, The Rabbit Counter Example, Injecting a Header Block into a SOAP Header, Configuring the Client-Side SOAP Handler, Adding a Handler Programmatically on the Client Side, Generating a Fault from a @WebMethod, Adding a Logical Handler for Client Robustness, Adding a Service-Side SOAP Handler, Summary of the Handler Methods, the Rabbit Counter As a SOAP 1.2 Service, The Message Context and Transport Headers, An Example to Illustrate Transport-Level Access, Web Services and Binary Data, Three Options for SOAP Attachments, Using Base64 Encoding for Binary Data, Using MTOM for Binary Data.

MODULE-III
RESTful Web Services **(25 Lectures)**

What Is REST? Verbs and Opaque Nouns, From @WebService to @WebServiceProvider, A RESTful Version of the Teams Service, The WebServiceProvider Annotation, Language Transparency and RESTful Services, Summary of the RESTful Features, Implementing the Remaining CRUD Operations, Java API for XML Processing, The Provider and Dispatch Twins, A Provider/Dispatch Example, More on the Dispatch Interface, A Dispatch Client Against a SOAP-based Service, Implementing RESTful Web Services As HttpServlets, The RabbitCounterServlet, Requests for MIME-Typed Responses, Java Clients Against Real-World RESTful Services, The Yahoo! News Service, The Amazon E-Commerce Service: REST Style, The RESTful Tumblr Service, WADLing with Java-Based RESTful Services, JAX-RS: WADLing Through Jersey, The Restlet Framework.

TEXT BOOKS:

1. *Java Web Services: Up and Running, Book by Martin Kalin*

REFERENCE BOOKS:

2. *Restful Java with Jax-RS by Bill Burke, Packtpblication.*
3. *Java Web Services in a Nutshell - O'Reilly Media by O'Reilly - O'Reilly Media.*
4. *Java Web Services for Experienced Programmers by DEITEL® publication.*
5. *Professional Java Web Services by Scott Cable (Author), Ben Galbraith (Author), RominIrani (Author), Mack Hendricks (Author).*

BSIT3602 Compiler Design

Code	*Course Type*	*Credits*	*L-T-P*
BSIT3602	***Theory + Practice***	***6***	***4-0-2***

Objective

- To introduce the major concept areas of language translation and compiler design.
- To enrich the knowledge in various phases of compiler ant its use

Learning outcome

- To apply the knowledge of lex tool & yacc tool to devleop a scanner & parser.
- To design & conduct experiments for Intermediate Code Generation in compiler.

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	Written examination
	Assignment	*NIL*	Report and Presentation
	Experiments	*30*	Lab work, report
	Project	*NIL*	Report and presentation
	Quiz	*NIL*	Surprise/preannounced ones
External Examination		*20*	Practice Examination

MODULE-I

1. Introduction: (6 Lectures)
Translator, types of translator, Compiler, types of compiler, Compiler Vs. Interpreter. Overview and phases of compilation.

2. Lexical Analysis: (6 Lectures)
Non-deterministic and deterministic finite automata (NFA & DFA), NFA Vs. DFA, regular grammar, regular expressions and regular languages, design of a lexical analyser as a DFA, lexical analyser generator.

3. Syntax Analysis: (9 Lectures)
Role of a parser, context free grammars and context free languages, parse trees andderivations, ambiguous grammar, Eliminate the left recursion and null production.

4. Top Down Parsing: (7 Lectures)
Recursive descent parsing, LL (1) grammars, non-recursive predictive parsing, error reporting and recovery.

MODULE-II

5. **Bottom Up Parsing: (10 Lectures)**
Handle pruning and shift reduces parsing, SLR parsers and construction of SLR parsing tables, LR(1) parsers and construction of LR(1) parsing tables, LALR parsers and construction of efficient LALR parsing tables, parsing using ambiguous grammars, error reporting and recovery, parser generator.

6. Syntax Directed Translation: (5 Lectures)
Syntax directed definitions (SDD), inherited and synthesized attributes, dependency graphs, evaluation orders for SDD, semantic rules, application of syntax directed translation.

7. Symbol Table: (4 Lectures)
Structure and features of symbol tables, symbol attributes and scopes.

MODULE-III

8. Intermediate Code Generation: (9 Lectures)
Introduction, benefits and types of intermediate code generation, three address codes - quadruples and triples, DAG for expressions, types and declarations, translation of expressions, translation of Boolean expressions and control flow statements, back patching, intermediate code generation for procedures.

9. Run Time Environment: (9 Lectures)
Storage organizations, static and dynamic storage allocations, stack allocation, Activation of the procedure and the activation record.

10. Code Generations: **(5 Lectures)**
Introduction, Major Issues of Code generation, registers allocation, simple codegeneration using basic blocks.

11. Elements of Code Optimization: **(5 Lectures)**
Objective, peephole optimization, redundant and un-reachable codes, concepts of elimination of local common sub-expressions, basics of flow of control optimization.

TEXT BOOKS:
1. Compilers - Principles, Techniques & Tools, Alfred V. Aho, Monica S. Lam, Ravi Sethi& Jeffrey D. Ullman Pearson

REFERENCE BOOKS:
1. Compiler Design, Author: O. G. Kakde , Publisher: Laxmi Publication.
2.Compiler Design, Authors: S K Sahu, A B Biswal, Publisher: Alok

Discipline Specific Elective-1

Nomenclature

Subject Name	**Code**	**Type of course**	**L-T-P**	**Prerequisite**
Cloud Computing	BSIT3609	Theory & Practice	4+0+3	Nil

Objective

- *Understanding the basic concepts of cloud computing and its implementation.*

Learning outcome

- *Have an idea about cloud architecture, Amazon cloud services, and Microsoft cloud services.*

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	Written examination
	Experiments	*30*	Lab work, Report and Viva Voce
External Examination	Semester	*30*	Written examination
	Practice (External)	*20*	Practice Exam with viva voce
Total		*100*	

MODULE- I: **(20Lectures)**

Introduction : Introduction to Cloud Computing, History and Evolution of Cloud Computing, Types of clouds, Private Public and hybrid clouds, Cloud Computing architecture, Cloud computing infrastructure, Merits of Cloud computing, , Cloud computing delivery models and services (IaaS, PaaS, SaaS), obstacles for cloud technology, Cloud vulnerabilities, Cloud challenges, Practical applications of cloud computing.

MODULE- II: **(20 Lectures)**

Cloud Computing Companies and Migrating to Cloud : Web-based business services, Delivering Business Processes from the Cloud: Business process examples, Broad Approaches to Migrating into the Cloud, The Seven-

Step Model of Migration into a Cloud, Efficient Steps for migrating to cloud., Risks: Measuring and assessment of risks, Company concerns Risk Mitigation methodology for Cloud computing, Case Studies.

Cloud Cost Management and Selection of Cloud Provider :Assessing the Cloud: software Evaluation, System Testing, Seasonal or peak loading, Cost cutting and cost-benefit analysis, Selecting the right scalable application. Considerations for selecting cloud solution. Understanding Best Practices used in selection of Cloud service and providers, Clouding the Standards and Best Practices Issue: Interoperability, Portability, Integration, Security, Standards Organizations and Groups associated with Cloud Computing, Commercial and Business Consideration.

MODULE- III: (15 Lectures)

Governance in the Cloud: Industry Standards Organizations and Groups associated with Cloud Computing, Need for IT governance in cloud computing, Cloud Governance Solution: Access Controls, Financial Controls, Key Management and Encryption, Logging and Auditing, API integration. Legal Issues: Data Privacy and Security Issues, Cloud Contracting models, Jurisdictional Issues Raised by Virtualization and Data Location, Legal issues in Commercial and Business Considerations.

5 ten cloud do an do nots::Don't be reactive, do consider the cloud a financial issue, don't go alone, do think about your architecture, don't neglect governance, don't forget about business purpose, do make security the centerpiece of your strategy, don't apply the cloud to everything, don't forget about Service Management, do start with a pilot project.

Text Books:

1. Cloud Computing: Principles and Paradigms, RajkumarBuyya, James Broberg, Andrzej M. Goscinski,, John Wiley and Sons Publications, 2011

List of Programs:

1. Study the basic cloud architecture and represent it using a case study
2. Enlist Major difference between SAAS PAAS & Iaas also submit a research done on various companies in cloud business and the corresponding services provided by them , tag them under SAAS , Paas & Iaas.
3. Study and present a report on Jolly cloud.
4. Present a report on obstacles and vulnerabilities in cloud computing on generic level
5. Present a report on Amazon cloud services.
6. Present a report on Microsoft cloud services.
7. Present a report on cost management on cloud
8. Enlist and explain legal issues involved in the cloud with the lelp of a case study
9. Explain the process of migrating to cloud with a case study.
10. Present a report on google cloud and cloud services.

Discipline Specific Elective-2

Nomenclature

Subject Name	Code	Type of course	L-T-P	Prerequisite
Big Data Analytics	BSIT3610	Theory & Practice	4+0+3	Nil

Objective

- *Understanding the basic and advanced methods to big data technologies and tools*

Learning outcome

- *Have an idea about the tips and tricks applicable for Big Data use cases and its solutions.*
- *Able to apply Hadoop ecosystem components in real world scenarios.*

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	Written examination
	Experiments	*30*	Lab work, Report and Viva Voce
External Examination	Semester	*30*	Written examination
	Practice (External)	*20*	Practice Exam with viva voce
Total		*100*	

UNIT - I **(24 hours)**

INTRODUCTION TO BIG DATA

Introduction – distributed file system – Big Data and its importance, Four Vs, Drivers for Big data, Big data analytics, Big data applications. Algorithms using map reduce, Matrix-Vector Multiplication by Map Reduce.

INTRODUCTION HADOOP

Big Data – Apache Hadoop & Hadoop EcoSystem – Moving Data in and out of Hadoop – Understanding inputs and outputs of MapReduce - Data Serialization.

UNIT - II **(28 hours)**

NOSQL

NoSQL Datbases, SQL Versus NoSQL, Advantages and Disadvantages of NoSQL.

HADOOP ARCHITECTURE

Hadoop Architecture, Hadoop Storage: HDFS, Common Hadoop Shell commands, Anatomy of File Write and Read, Name Node, Secondary Name Node, and Data Node, Hadoop MapReduce paradigm, Map and Reduce tasks, Job, Task trackers - Cluster Setup – SSH & Hadoop Configuration – HDFS Administering – Monitoring & Maintenance.

UNIT - III **(28 hours)**

HIVE AND HIVEQL, HBASE

Hive Architecture and Installation, Comparison with Traditional Database, HiveQL - Querying Data - Sorting And Aggregating, Map Reduce Scripts, Joins & Subqueries, HBase concepts- Advanced Usage, Schema Design,

Advance Indexing - PIG, Zookeeper - how it helps in monitoring a cluster, HBase uses Zookeeper and how to Build Applications with Zookeeper.

REFERENCES

1. Boris lublinsky, Kevin t. Smith, Alexey Yakubovich, "Professional Hadoop Solutions", Wiley, ISBN: 9788126551071, 2015.
2. Seema Acharya, Subhasisni Chellappan, "Big Data And Analytics", Wiley, ISBN: 9788126554782, 2016.

Discipline Specific Elective-3

Nomenclature

Subject Name	Code	Type of course	L-T-P	Prerequisite
Network Programming	BSIT3611	Theory & Practice	4+0+3	Nil

Objective

- *Understand the basic principles of computer networks*
- *Practice the programming aspects of computer networks in Unix platform*

Learning outcome

- *Develop client/server communication application using socket programming*

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	Written examination
	Experiments	*30*	Lab work, Report and Viva Voce
External Examination	Semester	*30*	Written examination
	Practice (External)	*20*	Practice Exam with viva voce
Total		*100*	

Module I **(10 Hrs)**

Introduction to Network Programming : OSI model, Unix standards, TCP and UDP & TCP connection establishment and Format, Buffer sizes and limitation, standard internet services, Protocol usage by common internet application.

Sockets : Sockets Address structures, value – result arguments, Byte ordering and manipulation function and related functions

Module II **(15 Hrs)**

Elementary TCP Sockets – Socket, connect, bind, listen, accept, fork and exec function, concurrent servers. Close function and related function.

TCP Client/Server Example: Introduction, TCP Echo server functions, Normal startup, terminate and signal handling server process termination, Crashing and Rebooting of server host shutdown of server host.

Module III **(15 Hrs)**

Elementary UDP Sockets: Introduction UDP Echo server function, lost datagram, summary of UDP example, Lack of flow control with UDP, determining outgoing interface with UDP.

Elementary name and Address conversions: DNS, gethostbyname function, Resolvers and Name Servers, Resolver Functions, Other networking information.

TEXT BOOKS:

1. UNIX Network Programming, Vol. I, Sockets API, 2nd Edition. - W.Richard Stevens, Pearson Edn. Asia.
2. UNIX Network Programming, 1st Edition, - W.Richard Stevens. PHI.

REFERENCES:

1. UNIX Systems Programming using C++ T CHAN, PHI.
2. UNIX for Programmers and Users, 3rd Edition Graham GLASS, King abls, Pearson Education
3. Advanced UNIX Programming 2nd Edition M. J. ROCHKIND, Pearson Education

Possible Practice Problems

1. Program to implement a simple socket client using all socket handling function.
2. Program to implement a simple socket server using all socket handling function.
3. Program to read a text line from a descriptor, one byte at a time.
4. Program to create dynamic server that prints client IP address and port.
5. Program to create TCP echo server and TCP client server.
6. Program to create TCP client that establishes five connections with server.
7. Program to create TCP server that handles an error of EINTR from accept.
8. Program to create an unconnected UDP socket.
9. Program to create a connected UDP socket.
10. Program to create a protocol independent daytime server.

Discipline Specific Elective-4

Nomenclature

Subject Name	Code	Type of course	L-T-P	Prerequisite
System Programming	BSIT3612	Theory & Practice	4-0-3	Nil

Objective

- *Understand the basic design of various system software.*

Learning outcome

- ***Implement the execution process of HLL programs and the working of scanners and parsers***

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	Written examination
	Experiments	*30*	Lab work, Report and Viva Voce
External Examination	Semester	*30*	Written examination
	Practice (External)	*20*	Practice Exam with viva voce
Total		*100*	

Module I **(30 Hrs):**

System Software:Introduction and overview,Software Hierarchy,Systems Programming,Interfaces, Computer Languages and Tools, Life Cycle of a Source Program, System Software Development, Recent Trends in Software Development, Levels of System Software.

Programming Languages and Language Processors, Program Execution, Fundamental of Language Processing, Symbol Tables**.**

Module II **(35 Hrs):**

Assemblers:Assembly Language Programming,Design of the Assembler,Types of Assemblers, Two-Pass Assemblers and One-Pass Assemblers,Algorithm of Single Pass Assembler, Multi-Pass Assemblers, Advanced Assembly Process, Variants of Assemblers.

Linker,Loader:Introduction,Linking Concept,Design of a Linker,Linking in MSDOS, Overlay Structured Programs, Dynamic Linking, Loaders, Different Loading Schemes, Sequential and Direct Loaders, Compile-and-Go Loaders,Linkers v/s Loaders.

Module III **(25 Hrs):**

Compilers:Binding Times, Data Structure used in Compiling,Memory Allocation, Compilation of Expression,Compilation of Control Structure,Code Optimization.

Programming Language Grammars:Classification of Grammar, Ambiguity,Scanning, Parsing, Top Down Parsing, Bottom up Parsing, Language Processor Development Tools, LEX, YACC.

Practice:

List of Experiments may include the followings

1. Write a program to implement the lexical analyzer.
2. Write a Lexical Analyzer
3. Write a program to left factor the given grammar.
4. Write a program to remove the Left Recursion from a given grammar.
5. Implement Recursive Descendent Parsing for the given Grammar
6. Implement Predictive Parser for the given grammar

Reference

Text Books:

1. *W. Richard Stevens, Stephen A. Rago , Advanced Programming in the UNIX Environment , 3rd Edn,Addison-Wesley Professional, 2013*

2. *D M Dhamdhere ,System Programming McGraw Hill Publication*
3. *Leland L. Beck,System Software – An Introduction to Systems Programming,3rd Edition, Pearson Education Asia, 2000*
4. *Srimanta Pal ,System Programming, OXFORD Publication*
5. *Reference Books:*
6. *Mike Loukides and Andy Oram , Programming with GNU Software , O'Reilly*
7. *Bruce Molay,Understanding UNIX/LINUX Programming: A Guide to Theory and Practice, Prentice Hall, 2003*
8. *System Programming and Compiler Construction by R.K. Maurya & A. Godbole.*

Discipline Specific Elective-5

Nomenclature

Subject Name	**Code**	**Type of course**	**L-T-P**	**Prerequisite**
Information Security	BSIT3613	Theory & Practice	4-0-3	Nil

Objective

- *Understand how information security attempts to overcome attack on sensitive information*

Learning outcome

- ***Develop,Implement Physical Security, Understand Security Baselines and Ensure Infrastructure and Network Security***

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	Written examination
	Experiments	*30*	Lab work, Report and Viva Voce
External Examination	Semester	*30*	Written examination
	Practice (External)	*20*	Practice Exam with viva voce
Total		*100*	

Module I **(30 Hrs):**Introduction: Security Principles, CIA, Threats, Attacks, Security Types: Software Security-Vulnerabilities and protections. Practical Cryptography: Encryption, authentication, hashing, symmetric and asymmetric cryptography, Digital Signatures and Certificates. Network Security: Network security issues, Sniffing, IP spoofing, E-Mail security and Architecture, IPSec, SSL, PGP,Virus, Worms, Trojan.

Module II **(35 Hrs):**

Firewalls: Requirements,features of firewall, Types of firewall. Intrusion Detection Systems and Security Baselines. Cyber Security:Cyber Crime and security, Security tools, Introduction to Digital Forensic,fingerprinting, TCP/IP stack masking, Social Engineering.

Module III **(25 Hrs):**

Web Security:Web application Security, Privacy and Anonymity,policy. Data compression,Huffman coding. Computer Forensics and the Law. Privacy Issues,Risk Management.

Practice:

Lab work will be on the setting up of network firewalls, Intruder Detection systems, security and performance analysis of cryptography algorithms and digital signature algorithms.

Reference

Text Books:

1. *William Stallings, Lawrie Brown, Computer Security: Principles and Practice,6th Edn.*
2. *Behrouz A. Forouzan,Cryptography and Network Security,3rd Edn,2015*

Reference Books:

1. *Niels Ferguson,Cryptography Engineering: Design Principles and Practical Applications,1st Edn*
2. *Matt Bishop, Introduction to Computer Security, Addison-Wesley*

Discipline Specific Elective-6

Nomenclature

Subject Name	**Code**	**Type of course**	**L-T-P**	**Prerequisite**
Data Mining	BSIT3614	Theory & Practice	4-0-3	Nil

Objective

- Provide students with an overview of the methodologies and approaches to datamining
- Provide the students with practice on applying data mining solutions using commondata mining software tool (e.g. WEKA, SPSS, Data Miner)
- Enhance students communication and problem solving skills

Learning outcome

- Able to understand the difference between Data Warehousing and generaldatabases.
- Determine the different steps followed in Data mining and pre-processing for Data mining.
- Are able to apply Association Rule mining.
- Are familiar with a data mining software system and use it forsolving data mining problems

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Midterm Test	*20*	Written examination
	Assignment		Report and Presentation
	Experiments	*30*	Practice work, report
	Project		Report and presentation
	Quiz		Surprise/preannounced ones
External Examination		*30*	Written examination
		20	Practice Exam with viva voce
Total		*100*	

MODULE-I

Introductionto Data Mining: **(6 Lectures)**

Data mining, Related technologies - Machine Learning, DBMS, OLAP, Statistics, Data Mining Goals , Stages of the Data Mining Process, Data Mining Techniques, Knowledge Representation Methods, Applications, Example: weather data.

Data Warehouse and OLAP: **(5 Lectures)**

Data Warehouse and DBMS, Multidimensional data model, OLAP operations, Example: loan data set

Data preprocessing : **(5 Lectures)**

Data cleaning, Data transformation, Data reduction, Discretization and generating concept hierarchies, Installing Weka 3 Data Mining System, Experiments withWeka - filters, discretization, Experiments with Weka - visualization

MODULE-II

Attribute-oriented analysis: **(6 Lectures)**

Attribute generalization, Attribute relevance, Class comparison, Statistical measures, Experiments with Weka - using filters and statistics

Data mining algorithms: Association rules: **(8 Lectures)**

Motivation and terminology, Example: mining weather data, Basic idea: item sets, Generating item sets and rules efficiently, Correlation analysis, Experiments with Weka - mining association rules

Data mining algorithms: Classification : **(6 Lectures)**

Basic learning/mining tasks, Inferring rudimentary rules: 1R algorithm, Decision trees, Covering rules, Experiments with Weka - decision trees, rules

MODULE-III

Data mining algorithms: Prediction : **(10 Lectures)**

The prediction task, Statistical (Bayesian) classification, Bayesian networks, Instance-based methods (nearest neighbor), Linear models, Experiments with Weka - Prediction

Clustering : **(14 Lectures)**

Basic issues in clustering, First conceptual clustering system: Cluster/2, partitioning methods: k-means, expectation maximization (EM), Hierarchical methods: distance-based agglomerative and divisible clustering, Conceptual clustering: Cobweb, Experiments with Weka - k-means, EM, Cobweb

Practice:

1. Creation of a Data Warehouse.
2. Apriori Algorithm.
3. FP-Growth Algorithm.
4. K-means clustering.
5. One Hierarchical clustering algorithm.
6. Bayesian Classification.
7. Decision Tree.
8. Support Vector Machines.
9. Applications of classification for web mining.
10. Case Study on Text Mining or any commercial application.

TEXT BOOKS:

1. Jiawei Han and MichelineKamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 3rd edition, 20.

REFERENCE BOOKS:

1. Ian H. Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques withJava Implementations, Morgan Kaufmann Publishers, 2nd edition, 2005.

Discipline Specific Elective-7

Nomenclature

Subject Name	**Code**	**Type of course**	**L-T-P**	**Prerequisite**
Advanced Internet Security	BSIT3615	Theory & Practice	4-0-3	Nil

Objective

- The course is designed to provide exposure to internet security threats, vulnerabilities of protocols and the different types of attacks covering preventive and defensive mechanisms, intrusion detection techniques, and tracing the source of attacks.

Learning outcome

- Assess Internet security threats and estimate the economic impact of network attacks
- Understand different Internet attacks and the vulnerabilities in internetworking as well as identify potential vulnerabilities in internets with idea of how to resolve identified vulnerabilities
- Describe tools used for the prevention and detection of Internet attacks as well as trace the source of some types of Internet attacks
- Conduct, document and present a literature review on a topic related to Internet security

Evaluation Systems

Internal Examination	***Component***	***% of Marks***	***Method of Assessment***
	Internal Test – I, II, III	*20*	Written Examination

	Experiments	*30 + 20*	Lab Work Report
External Examination		*30*	
Total		*100*	

Module I **(35 Hrs)**

Internet Security Introduction and Overview: Introduction to Information Security – What is CIA? Difference between Internet Security and Information Security **Internet Security Concepts**: Objectives, Roles, Domains **Advanced Networking**: Information Security Concepts and Terminologies, Risk Management and Control, Common Attack Types and Vectors, Internet Security Policies , Processes and Procedures, **Internet Security Controls** : People, Process and Technology

Module II **(33 Hrs)**

Security Architecture Principles: Overview of Security Architecture, The ISO/OSI Model, Defense-In-Depth, Isolation and Segmentation Procedures, **Monitoring Detection and Logging**, Security of Networks, **Vulnerabilities and Patch Management**, Security Penetration Testing, **Network Security Basics**: Operating System Basics, Application Security Basics, Data Security Basics and Laws

Module III (**22** Hrs)

IS Incident Management: Event vs. Incident , Security Incident Response & Treatment , Investigations, **Legal Holds and Preservation of Evidences**, Forensics & Investigation Basics , Information Security Management Systems Standard ,IS Global Standards: Business Continuity Management System Standard, IT Disaster recovery Best Practice , Risk Management standard , Data Privacy Standard , Crisis Management Standard , Quality management Standard.

References:

1. Computer networks: Tanenbaum, Andrew S, Prentice Hall
2. Data communication & networking: Forouzan, B. A.
3. Computer network protocol standard and interface Uyless, Black