

Learning Outcomes Based Curriculum Framework (LOCF)
for
Computer Science & Information Technology

Undergraduate Programme
2020

Department of Computer Science & Information Technology
Cotton University
Panbazar, Guwahati
Assam

PART I

1.1 Introduction

“Computers are incredibly fast, accurate, and stupid. Human beings are incredibly slow, inaccurate, and brilliant. Together they are powerful beyond imagination.” - Albert Einstein

The study of computing, automation, and information is known as computer science. Theoretical disciplines (such as algorithms, theory of computing, and information theory) and practical disciplines (such as programming) are all part of computer science (including the design and implementation of hardware and software). Computer science is distinct from computer programming in that it is a field of academic research.

Computer science is based on algorithms and data structures. The theory of computing is concerned with abstract models of computation and the general types of problems that they may solve. The areas of cryptography and computer security are concerned with the development of methods for secure communication and the prevention of security flaws. Image generation is addressed by computer graphics and computational geometry. Database theory is concerned with the administration of data repositories, whereas programming language theory is concerned with methods to the representation of computer operations. Human–computer interaction studies the interactions between humans and computers, whereas software engineering studies the design and concepts of software development. Operating systems, networks, and embedded systems are all research areas that look at the concepts and design of complex systems. The building of computer components and computer-operated equipment is referred to as computer architecture. Artificial intelligence and machine learning aspire to synthesize goal-oriented processes present in people and animals, such as problem-solving, decision-making, environmental adaptability, planning, and learning. Computer vision tries to comprehend and process image and video data, whereas natural-language processing strives to understand and process textual and linguistic data in artificial intelligence.

This undergraduate programme in computer science and information technology is designed to introduce the basic concepts of the subject and to enable the students to understand and analyze the current-edge technologies, issues and challenges through the various applications of computer science. The programme covers fundamental concepts of the subject - the foundation subjects, core subjects, department specific elective subjects, and skill enhancement courses. The programme is designed in such a manner that it enables the students to apply core and programming knowledge to solve a wide range of real-life problems and issues and acquire research skills to produce research findings using in-depth subject knowledge, statistical tools, programming skills and current-edge technologies.

1.2 Learning Outcomes-based Approach to Curriculum Planning and Development

The basic objective of the learning outcome based approach to curriculum planning and development is to focus on demonstrated achievement of outcomes (expressed in terms of knowledge, understanding, skills, attitudes and values) and academic standards expected of graduates of a programme of study. Learning outcomes specify what graduates completing a particular programme of study are expected to know, understand and be able to do at the end of their programme of study.

The expected learning outcomes are used to set the benchmark to formulate the course outcomes, programme specific outcomes, programme outcomes and graduate attributes. These outcomes are essential for curriculum planning and development, and in the design, delivery and review of academic programmes. They provide general direction and guidance to the teaching-learning process and assessment of student learning levels under a specific programme.

The overall objectives of the learning outcomes-based curriculum framework are to:

- Help formulate graduate attributes, qualification descriptors, programme learning outcomes and course learning outcomes that are expected to be demonstrated by the holder of a qualification;
- Enable prospective students, parents, employers and others to understand the nature and level of learning outcomes (knowledge, skills, attitudes and values) or attributes a graduate of a programme should be capable of demonstrating on successful completion of the programme of study;
- Maintain national standards and international comparability of learning outcomes and academic standards to ensure global competitiveness, and to facilitate student/graduate mobility;
- Provide higher education institutions an important point of reference for designing teaching-learning strategies, assessing student learning levels, and periodic review of programmes and academic standards.

1.3 Key outcomes underpinning curriculum planning and development

The learning outcomes-based curriculum framework is a framework based on the expected learning outcomes and academic standards that are expected to be attained by graduates of a programme of study. The key outcomes that underpin curriculum planning and development include Graduate Attributes, Programme Outcomes, Programme Specific Outcomes, and Course Outcomes.

1.3.1 Graduate Attributes

The disciplinary expertise or technical knowledge that has formed the core of the university courses. They are qualities that also prepare graduates as agents for social good in future. Some of the characteristic attributes that a graduate should demonstrate are as follows:

1. **Disciplinary knowledge:** Capable of demonstrating comprehensive knowledge and understanding of one or more disciplines.
2. **Research-related skills:** A sense of inquiry and capability for asking relevant/appropriate questions, problematising, synthesizing and articulating.
3. **Analytical reasoning:** Ability to evaluate the reliability and relevance of evidence; identify logical flaws and holes in the arguments of others.
4. **Critical thinking:** Capability to apply analytic thought to a body of knowledge.
5. **Problem solving:** Capacity to extrapolate from what one has learned and apply their competencies to solve different kinds of non-familiar problems.
6. **Communication Skills:** Ability to express thoughts and ideas effectively in writing and orally.
7. **Information/digital literacy:** Capability to use ICT in a variety of learning situations, demonstrate ability to access, evaluate, and use a variety of relevant information sources; and use appropriate software for analysis of data.
8. **Self-directed learning:** Ability to work independently, identify appropriate resources required for a project, and manage a project through to completion.
9. **Cooperation/Team-work:** Ability to work effectively and respectfully with diverse teams.
10. **Scientific reasoning:** Ability to analyze, interpret and draw conclusions from quantitative/qualitative data; and critically evaluate ideas, evidence and experiences from an open-minded and reasoned perspective.
11. **Reflective thinking:** Critical sensibility to lived experiences, with self-awareness and reflexivity of both self and society.
12. **Multicultural competence:** Possess knowledge of the values and beliefs of multiple cultures and a global perspective.
13. **Moral and ethical awareness/reasoning:** Ability to embrace moral/ethical values in conducting one's life, formulate a position/argument about an ethical issue from multiple perspectives, and use ethical practices in all work.
14. **Leadership readiness/qualities:** Capability for mapping out the tasks of a team or an organization, and setting direction, formulating an inspiring vision, building a team who can help achieve the vision, motivating and inspiring team members to engage with that vision, and using management skills to guide people to the right destination, in a smooth and efficient way.
15. **Lifelong learning:** Ability to acquire knowledge and skills, including 'learning how to learn', that are necessary for participating in learning activities throughout life, through self-paced and self-directed learning aimed at personal development, meeting economic,

social and cultural objectives, and adapting to changing trades and demands of work place through knowledge/skill development/reskilling.

1.3.2 Programme Outcomes (POs) for Undergraduate programme

POs are statements that describe what the students graduating from any of the educational programmes should be able to do. They are the indicators of what knowledge, skills and attitudes a graduate should have at the time of graduation.

1. **In-depth knowledge:** Understand the concepts and processes related to an academic field of study and demonstrate the applicability of their domain knowledge and its links to related disciplinary areas/subjects of study.
2. **Specialized knowledge and skills:** Demonstrate procedural knowledge and skills in areas related to one's specialization and current developments, including a critical understanding of the latest developments in the area of specialization, and an ability to use established techniques of analysis and enquiry within the area of specialization.
3. **Analytical and critical thinking:** Demonstrate independent learning, analytical and critical thinking of a wide range of ideas and complex problems and issues.
4. **Research and Innovation:** Demonstrate comprehensive knowledge about current research in the subject of specialization; critical observation to identify research problems and to collect relevant data from a wide range of sources, analysis and interpretation of data using methodologies as appropriate to the area of specialization for formulating evidence-based research output.
5. **Interdisciplinary Perspective:** Commitment to intellectual openness and developing understanding beyond subject domains.
6. **Communication Competence:** Demonstrate effective oral and written communication skills to convey disciplinary knowledge and to communicate the results of studies undertaken in an academic field accurately in a range of different contexts using the main concepts, constructs and techniques of the subject(s) of study
7. **Career development:** Show proficiency in academic, professional, soft skills and employability required for higher education and placements.
8. **Teamwork:** Work in teams with enhanced interpersonal skills and leadership qualities.
9. **Commitment to the society and the Nation:** Recognise the importance of social, environmental, human and other critical issues faced by humanity at the local, national and international level; appreciate the pluralistic national culture and the importance of national integration.

1.3.3 Programme Specific Outcomes (PSOs) in Computer Science & Information Technology

Programme specific outcomes include subject-specific skills and generic skills, including transferable global skills and competencies, the achievement of which the students of a specific programme of study should be able to demonstrate for the award of the degree. The programme specific outcomes would also focus on knowledge and skills that prepare students for further study, employment, and citizenship. They help ensure comparability of learning levels and academic standards across universities and provide a broad picture of the level of competence of graduates of a given programme of study. The attainment of PSOs for a programme is computed by accumulating PSO attainment in all the courses comprising the programme.

PSO1: Ability to Identify and critically analyze real word problems in order to provide efficient solutions in the field of computer science and its allied branches.

PSO2: Ability to identify the role of computing while designing solutions for the problems, and anticipate the challenges as well as impact of the solution.

PSO3: Ability to apply domain specific knowledge in the development of useful ICT products/processes/ applications/models, which could help in the advancement of society/country at large.

PSO4: Ability to identify innovative solutions for the existing problems with the help of sound mathematical foundations, algorithmic principles, theoretical computer science and research-based knowledge.

PSO5: Ability to acquire domain specific expertise through discipline specific elective, general elective and skill enhancement courses as well as project works.

1.3.4 Course Level Learning Outcome Matrix

Course Level Learning Outcomes Matrix – Core Course

Programme Specific Outcomes	101	102	201	202	301	302	303	401	402	403	501	502	601	602
PSO1	X	X	X	X	X	X	X	X	X	X		X	X	X
PSO2	X	X	X			X		X	X	X		X		
PSO3						X	X		X	X	X		X	X
PSO4		X		X	X	X	X	X		X	X	X	X	

The department of Computer Science & Information Technology, Cotton University has student-centric teaching-learning pedagogies to enhance the learning experiences of the students. All classroom lectures are interactive using ICT-enabled techniques, allowing the students to have meaningful discussions, question and answer sessions. Apart from the physical classes, lectures are also held in online mode where students can have doubt clearing and discussions with the teachers. Most of the teachers use ICT facilities with power-point presentations, e-learning platforms and other innovative e-content platforms for student-centric learning methods. Apart from these, special lectures by invited experts, workshops, and seminars are held to augment knowledge, encourage innovative ideas and expose the students to global academic and research advancement.

The short-term projects, research projects, and assignments, which are the integral components of all the courses, enable the students to solve practical problems. Students are also being engaged in the in-house and external research projects for acquiring experiential learning. The laboratories of the department offer hands-on learning experiences to the students.

1.5 Assessment methods

A variety of assessment methods that are appropriate to the discipline are used to assess progress towards the course/programme learning outcomes. Priority is accorded to formative assessment. Progress towards achievement of learning outcomes is assessed using the following: closed-book examinations; problem based assignments; practical assignment; laboratory reports; individual project reports (case-study reports); team project reports; oral presentations, including seminar presentation; viva voce interviews; computerized testing and any other pedagogic approaches as per the context.

PART II

Structure of Under-Graduate programme in Computer Science & Information Technology

Outline of the courses under Choice Based Credit System:

Each course of a program will be of one of the following categories-

1. Core Course: A course, which should compulsorily be studied by a candidate as a core requirement is termed as a Core course.

2. Elective Course: Generally a course which can be chosen from a pool of courses and which may be very specific or specialized or advanced or supportive to the discipline/subject of study or which provides an extended scope or which enables an exposure to some other discipline/subject/domain or nurtures the candidate's proficiency/skill is called an Elective Course.

2.1 Discipline Specific Elective (DSE) Course: Elective courses may be offered by the main discipline/subject of study referred to as Discipline Specific Elective. The University may also offer discipline related Elective courses of interdisciplinary nature (to be offered by main discipline/subject of study).

2.2 Dissertation/Project: An elective course designed to acquire special/advanced knowledge, such as supplement study/support study to a project work, and a candidate studies such a course on his own with an advisory support by a teacher/faculty member is called dissertation/project.

3. Ability Enhancement Courses (AEC): The Ability Enhancement (AE) Courses may be of two kinds: Ability Enhancement Compulsory Courses (AECC) and Skill Enhancement Courses (SEC). "AECC" courses are the courses based upon the content that leads to Knowledge enhancement. These are mandatory for all disciplines. SEC courses are value-based and/or skill-based and are aimed at providing hands-on-training, competencies, skills, etc.

3.1 Ability Enhancement Compulsory Courses (AECC): Environmental Science, English Communication/MIL Communication.

3.2 Skill Enhancement Courses (SEC): These courses may be chosen from a pool of courses designed to provide value-based and/or skill-based knowledge.

Introducing Research Component in Under-Graduate Courses:

Project work/Dissertation is considered as a special course involving application of knowledge in solving / analyzing /exploring a real life situation / difficult problem. A Project/Dissertation work would be of 6 credits. A Project/Dissertation work may be given in lieu of a discipline specific elective paper.

Details of courses and corresponding Credits under Undergraduate (Honours):

Minimum credits required for the complete programme are:

(14 Core papers × 6 credit each) + (2 AECC papers × 2 credit each) + (2 SEC papers × 2 credit each) + (8 DSE papers × 6 credits each) = 140 Credits

Course		No. of Papers	Credits	
			Theory + Practical	Theory + Tutorial
I. Core Course		14	$14 \times (4+2) = 84$	$14 \times (5+1) = 84$
II. Elective Course Total : 8 Papers	Discipline Specific Elective	8	$8 \times (4+2) = 48$	$8 \times (5+1) = 48$
III. Ability Enhancement Courses (Min 2, Max. 4)	A. Ability Enhancement Compulsory Courses (AECC)	2	$2 \times (2+0) = 4$	$2 \times (2+0) = 4$
	B. Skill Enhancement Courses (SEC)	2	$2 \times (2+0) = 4$	$2 \times (2+0) = 4$
		26	Total credit= 140	Total credit= 140

*Wherever there is a practical there will be no tutorial and vice-versa.

* Figures in the parenthesis indicate the credits

Semester wise Course and Credit Distribution:

Course : Undergraduate (Honours)

Minimum credits required for the complete programme are:

$(14 \text{ Core papers} \times 6 \text{ credit each}) + (2 \text{ AECC papers} \times 2 \text{ credit each}) + (2 \text{ SEC papers} \times 2 \text{ credit each}) + (8 \text{ DSE papers} \times 6 \text{ credits each}) = 140 \text{ Credits}$

Semester	CORE COURSE No of Paper -14	Ability Enhancement Compulsory Course (AECC) No of Paper – 2	Skill Enhanceme nt Course (SEC) No of Paper - 2	Elective: Discipline Specific DSE No of Paper – 4
I	C 1 Credit- 6	*English Communication/ MIL for BA Course **Environmental Science for BCA Course Credit- 2		DSE-1 Credit- 6
	C 2 Credit- 6			
II	C 3 Credit- 6	*English Communication/ MIL for BCA Course **Environmental Science for BA Course Credit- 2		DSE-2 Credit- 6
	C 4 Credit- 6			
III	C 5 Credit- 6		SEC -1 Credit- 2	DSE-3 Credit- 6
	C 6 Credit- 6			
	C 7 Credit- 6			
IV	C 8		SEC -2	DSE-4

	Credit- 6		Credit- 2	Credit- 6
	C 9 Credit- 6			
	C 10 Credit- 6			
V	C 11 Credit- 6			DSE-5 Credit- 6
	C 12 Credit- 6			DSE -6 Credit- 6
VI	C 13 Credit- 6			DSE -7 Credit- 6
	C 14 Credit- 6			DSE -8 Credit- 6

Choice of Ability Enhancement Compulsory Course (AECC):

1. Students of BCA:

For First Semester-

*It is mandatory to take Environmental Science by all students.

For Second Semester-

*Students having an MIL course in the qualifying examination, will have to take the same MIL as an AECC course otherwise they will take English Communication as an AECC course.

2. Students of BA (Honours) :

For First Semester-

(a)Students with an MIL as their core subject: This category of students will take English Communication as their AECC course.

(b) Students with English as their core subject: They will take a MIL as their AECC course if that MIL was a subject for them in the qualifying examination, otherwise they will take English Communication as AECC course.

(c) Students with neither an MIL nor English as their core subject: They will take an MIL as their AECC course if that MIL was a subject for them in the qualifying examination, otherwise they will take English Communication AECC course.

For Second Semester-

*It is mandatory to take Environmental Science by all students.

COURSE STRUCTURE OF COMPUTER APPLICATION (UNDERGRADUATE PROGRAMME)

- 1 credit = 1 theory period of 1 hour duration per week (L)
- 1 credit = 1 tutorial period of 1 hour duration per week (T)
- 1 credit = 1 practical period of 2 hour duration per week (P)

There will be total 15 weeks class in one semester

Core		
Paper Code	Paper Title	L+T+P
Semester I		
BCA101C	Programming Fundamentals Using C/C++	4+0+2
BCA102C	Computer System Architecture	4+0+2
Semester II		
BCA201C	Programming in Java	4+0+2
BCA202C	Discrete Structure	4+0+2
Semester III		
BCA301C	Data Structures	4+0+2

BCA302C	Operating Systems	4+0+2
BCA303C	Computer Networks	4+0+2
Semester IV		
BCA401C	Design and Analysis of Algorithm	4+0+2
BCA402C	Software Engineering	4+0+2
BCA403C	Database Management System	4+0+2
Semester V		
BCA501C	Internet Technologies	4+0+2
BCA502C	Theory of Computation	5+1+0
Semester VI		
BCA601C	Artificial Intelligence	4+0+2
BCA602C	Computer Graphics	4+0+2

Discipline Specific Elective

DSE01	Computer Fundamentals	4+0+2
DSE02	Introduction to Database Systems	4+0+2
DSE03	Introduction to Programming	4+0+2
DSE04	Computer Networks and Internet Technologies	4+0+2
DSE05	Multimedia and Applications	4+0+2
DSE06	Programming in Python	4+0+2

DSE07	Programming in Visual Basic / Gambas	4+0+2
DSE08	Information Security and Cyber Laws	4+0+2
DSE09	Web and E-Commerce Technologies	4+0+2
DSE10	Information Security	4+0+2
DSE11	Network Programming	4+0+2
DSE12	Microprocessor	4+0+2
DSE13	Computational Linguistics	4+0+2
DSE14	Digital Image Processing	4+0+2
DSE15	Machine Learning	4+0+2
DSE16	Introduction to Data Sciences	4+0+2
DSE17	Cloud Computing	4+0+2
DSE18	Numerical Methods	4+0+2
DSE19	System Programming	4+0+2
DSE20	Operational Research	4+0+2
DSE21	Data Mining	4+0+2
DSE22	Project Work / Dissertation	4+0+2
DSE23	Big Data Analytics	4+0+2
DSE24	Soft Computing	4+0+2

Skill Enhancement Course

SEC01	Android Programming	1+0+1
SEC02	Programming in MATLAB	1+0+1
SEC03	HTML Programming	1+0+1
SEC04	XML Programming	1+0+1
SEC05	Oracle (SQL/PL-SQL)	1+0+1

SEC06	Programming in Python	1+0+1
SEC07	UNIX/LINUX Programming	1+0+1

Core Courses

Paper Code: BCA101C

Paper Title: Programming Fundamentals Using C/C++ (6 Credits, L+T+P = 4+0+2)

This course will teach you how to use the C++ programming language to learn structured and object-oriented programming. The course not only covers basic C++ structures, but it also delves into object-oriented programming concepts such as encapsulation, abstraction, inheritance, and polymorphism to write more efficient programmes.

On successful completion of the course, students will be able to:

1. Explain the significance of object-oriented programming.
2. Learn about the strategies of writing efficient and well-structured computer algorithms/programs.
3. Learn array processing algorithms coupled with iterative methods.
4. Solve programming problems using object oriented features.
5. Reuse classes to create new classes.
6. Handle external files as well as exceptions.

Unit	Content	No. of Lectures
1	Introduction to C and C++: History of C and C++, Overview of Procedural Programming and Object-Oriented Programming, Using main() function, Compiling and Executing Simple Programs in C++.	5
2	Data Types, Variables, Constants, Operators and Basic I/O: 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, putchar), Formatted and Console I/O (printf(), scanf(), cin, cout), Using Basic Header Files (stdio.h, iostream.h, conio.h etc).	6
3	Expressions, Conditional Statements and Iterative Statements: 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)	6
4	Functions and Arrays: 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	10
5	Derived Data Types (Structures and Unions): 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.	3
6	Pointers and References in C++: Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Pointers to structures, Problems with Pointers, 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
7	Memory Allocation in C++: 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	3
8	File I/O, Pre-processor Directives: 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	4

9	Using Classes in C++: 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.	7
10	Overview of Function Overloading and Operator Overloading: 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)	5
11	Inheritance, Polymorphism and Exception Handling: 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, Re-throwing exceptions.	8

Practical:

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 $S = 1 + 1/2 + 1/3 + 1/4 + \dots$.
4. WAP to compute the sum of the first n terms of the following series $S = 1 - 2 + 3 - 4 + 5 - \dots$.
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by the 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 the 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 an 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 the sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions or new operators.
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

2 Credits

20. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.
21. Create a Matrix class using templates. Write a menu-driven program to perform following Matrix operations (2-D array implementation): a) Sum, b) Difference, c) Product, and 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 areas. Overload assignment operator and equality operator.
24. Create a class Box containing length, breadth 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, breadth 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 a file created in the previous question and print it in the 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.
29. Write a program that will read 10 integers from the user and store them in an array. Implement an array using pointers. The program will print the array elements in ascending and descending order.

Unit	Content	No. of Lectures
1	Introduction to C and C++: History of C and C++, Overview of Procedural Programming and Object-Oriented Programming, Using main() function, Compiling and Executing Simple Programs in C++.	5
2	Data Types, Variables, Constants, Operators and Basic I/O: 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, putchar), Formatted and Console I/O (printf(), scanf(), cin, cout), Using Basic Header Files (stdio.h, iostream.h, conio.h etc).	6
3	Expressions, Conditional Statements and Iterative Statements: 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)	6
4	Functions and Arrays: 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	10
5	Derived Data Types (Structures and Unions): 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.	3
6	Pointers and References in C++: Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Pointers to structures, Problems with Pointers, 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
7	Memory Allocation in C++: 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	3
8	File I/O, Pre-processor Directives: 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	4
9	Using Classes in C++: 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.	7
10	Overloading of Functions, Operator Overloading, Operator Overloading, Namespace, Operator Overloading	5

Reading list:

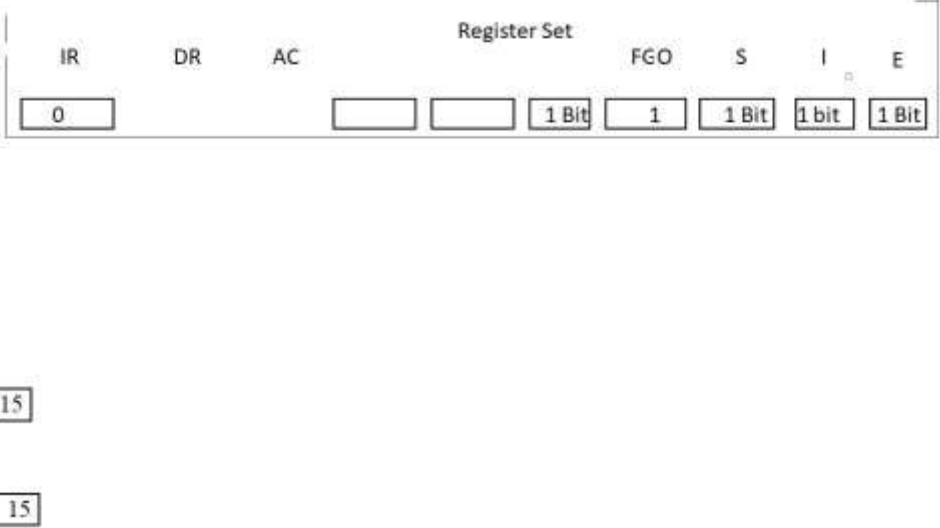
- Herbtz Schildt, "C++: The Complete Reference", Fourth Edition, McGraw Hill, 2003
- Bjarne Stroustrup, "The C++ Programming Language", 4th Edition, Addison-Wesley, 2013.
- Bjarne Stroustrup, "Programming--Principles and Practice using C++", 2nd Edition, Addison-Wesley 2014.
- E Balaguruswamy, "Object Oriented Programming with C++", Tata McGraw-Hill Education, 2008.
- Paul Deitel, Harvey Deitel, "C++ How to Program", 8th Edition, Prentice Hall, 2011.
- John R. Hubbard, "Programming with C++", Schaum's Series, 2nd Edition, 2000.
- Andrew Koeni, Barbara E. Moo, "Accelerated C++", Published by Addison-Wesley, 2000.
- Scott Meyers, "Effective C++", 3rd Edition, Published by Addison-Wesley, 2005.
- Harry, H. Chaudhary, "Head First C++ Programming: The Definitive Beginner's Guide", First Create space Inc, O-D Publishing, LLC USA, 2014
- Walter Savitch, "Problem Solving with C++", Pearson Education, 2007.
- Stanley B. Lippman, Josee Lajoie, Barbara E. Moo, "C++ Primer", Published by Addison Wesley, 5th Edition, 2012

Paper Code: BCA102C**Paper Title: Computer System Architecture (6 Credits, L+T+P = 4+0+2)**

The essential ideas of digital computer organization, design, and architecture are covered in this course. Its goal is to give students a fundamental grasp of the components that make up a computer system, as well as how these components are put together to create a digital computer system.

On successful completion of the course, students will be able to:

1. To make students understand the basic structure, operation and characteristics of digital computers.
2. Design Combinational Circuits using basic building blocks. Simplify these circuits using Boolean algebra and Karnaugh maps. Differentiate between combinational circuits and sequential circuits.
3. Represent data in binary form, convert numeric data between different number systems and perform arithmetic operations in binary.
4. To familiarize the students with hierarchical memory systems including cache memories and virtual memory.
5. To make students know the different ways of communicating with I/O devices and standard I/O interfaces.

Unit	Content	No. of Lectures
1	Introduction: Logic gates, boolean algebra, combinational circuits, circuit simplification, flip-flops and sequential circuits, decoders, multiplexers, registers, counters and memory units.	10
2	Data Representation and Basic Computer Arithmetic: Number systems, complements, fixed and floating point representation, character representation, addition, subtraction, magnitude-comparison, multiplication, and division algorithms for integers	12
3	Basic Computer Organization and Design: 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.	13
4	Central Processing Unit: 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.	15
5	Memory Organization: Cache memory, Associative memory, mapping, replacement algorithms, write policies.	6
6	Input-Output Organization: Input / Output: External Devices, I/O Modules, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access, I/O Channels.	8
	<p>Practical:</p> <p>1. Create a machine based on the following</p>  <p>The diagram shows a Register Set with the following fields: IR (0), DR, AC, FGO (1 Bit), S (1 Bit), I (1 bit), and E (1 Bit). Below the diagram are two boxes labeled '0 15'.</p>	2 Credits

Memory 4096 words 16 bits per word	0	Instruction format 3 4	15
--	---	---------------------------	----

Opcode	Address
--------	---------

Basic Computer Instructions

Memory Reference		Register Reference		Input-Output	
Symbol	Hex	Symbol	Hex	Symbol	Hex
AND	0xxx	CLA	E800	INP	F80 0
ADD	2xxx	CLE	E400	OUT	F40 0
LDA	4xxx	CMA	E200	SKI	F20 0
STA	6xxx	CME	E100	SKO	F10 0
BUN	8xxx	CIR	E080	ION	F08 0
BSA	Axxx	CIL	E040	IOF	F04 0
ISZ	Cxxx	INC	E020		
AND_I	1xxx	SPA	E010		
ADD_I	3xxx	SNA	E008		
LDA_I	5xxx	SZA	E004		
STA_I	7xxx	SZE	E002		
BUN_I	9xxx	HLT	E001		
BSA_I	Bxxx				
ISZ_I	Dxxx				

Optional

2. Create the micro operations and associate with instructions as given in the chapter (except interrupts). Design the register set, memory and the instruction set. Use this machine for the assignments of this section.
3. Create a Fetch routine of the instruction cycle.
4. Simulate the machine to determine the contents of AC, E, PC, AR and IR registers in hexadecimal after the execution of each of following register reference instructions:

a. CLA	e. CIR	i. SNA
b. CLE	f. CIL	j. SZA
c. CMA	g. INC	k. SZE
d. CME	h. SPA	l. HLT

Paper Title: Programming in Java (6 Credits, L+T+P = 4+0+2)

The essential ideas of digital computer organization, design, and architecture are covered in this course. Its goal is to give students a fundamental grasp of the components that make up a computer system, as well as how these components are put together to create a digital computer system.

Course Learning Outcomes:

On successful completion of the course, students will be able to:

1. Knowledge of the structure and model of the Java programming language
2. Use the Java programming language for various programming technologies
3. Implement Exception Handling and File Handling
4. Implement multiple inheritance using Interfaces
5. Use AWT and Swing to design GUI applications

Unit	Content	No. of Lectures
1	Review of Object Oriented Programming and Java Fundamentals: Structure of Java programs, Classes and Objects, Data types, Type Casting, Looping Constructs.	10
2	Interfaces Interface basics; Defining, implementing and extending interfaces; Implementing multiple inheritance using interfaces Packages Basics of packages, Creating and accessing packages, System packages, Creating user defined packages	12
3	Exception handling using the main keywords of exception handling: try, catch, throw, throws and finally; Nested try, multiple catch statements, creating user defined exceptions	8
4	File Handling Byte Stream, Character Stream, File I/O Basics, File Operations	8
5	AWT and Event Handling: The AWT class hierarchy, Events, Event sources, Event classes, Event Listeners, Relationship between Event sources and Listeners, Delegation event model, Creating GUI applications using AWT.	12
6	Swing Introduction to Swing, Swing vs. AWT, Hierarchy for Swing components, Creating GUI applications using Swing	10
	Practical: <ol style="list-style-type: none">1. Design a class Complex having a real part (x) and an imaginary part (y). Provide methods to perform the following on complex numbers:<ul style="list-style-type: none">● Add two complex numbers.● Multiply two complex numbers.● toString() method to display complex numbers in the form: x + iy2. Create a class TwoDim which contains private members as x and y coordinates in package P1. Define the default constructor, a parameterized constructor and override toString() method to display the coordinates. Now reuse this class and in package P2 create another class ThreeDim, adding a new dimension as z as its private member. Define the constructors for the subclass and override toString() method in the subclass also. Write appropriate methods to show dynamic method dispatch. The main() function should be in a package P.	2 Credits

- 3.** Define an abstract class Shape in package P1. Inherit two more classes: Rectangle in package P2 and Circle in package P3. Write a program to ask the user for the type of shape and then using the concept of dynamic method dispatch, display the area of the appropriate subclass. Also write appropriate methods to read the data. The main() function should not be in any package.
- 4.** Create an exception subclass UnderAge, which prints “Under Age” along with the age value when an object of UnderAge class is printed in the catch statement. Write a class exceptionDemo in which the method test() throws UnderAge exception if the variable age passed to it as argument is less than 18. Write main() method also to show working of the program.
- 5.** Write a program to implement stack. Use exception handling to manage underflow and overflow conditions.
- 6.** Write a program that copies content of one file to another. Pass the names of the files through command-line arguments.
- 7.** Write a program to read a file and display only those lines that have the first two characters as '/' (Use try with resources).
- 8.** Write a program to create an Applet. Create a frame as a child of applet. Implement mouseClicked(), mouseEntered() and mouseExited() events for applet. Frame is visible when the mouse enters the applet window and hidden when the mouse exits from the applet window.
- 9.** Write a program to display a string in a frame window with pink color as background.
- 10.** Write a program to create an Applet that has two buttons named “Red” and “Blue”. When a button is pressed the background color of the applet is set to the color named by the button’s label.
- 11.** Create an applet which responds to KEY_TYPED events and updates the status window with a message (“Typed character is: X”). Use adapter class for other two events.
- 12.** Create an applet with two buttons labeled ‘A’ and ‘B’. When button ‘A’ is pressed, it displays your personal information (Name, Course, Roll No, College) and when button ‘B’ is pressed, it displays your CGPA from the previous semester.
- 13.** Write a program that creates a Banner and then creates a thread to scroll the message in the banner from left to right across the applet’s window.
- 14.** Rewrite the applet programs using Swing.

Reading Books:

- Schildt, H. (2018). Java: The Complete Reference. 10th edition. McGraw-Hill Education. Additional Resources:
- Balaguruswamy E. (2014). Programming with JAVA: A Primer. 5th edition. India: McGraw Hill Education
- Horstmann, C. S. (2017). Core Java - Vol. I – Fundamentals (Vol. 10). Pearson Education
- Schildt, H., & Skrien, D. (2012). Java Fundamentals - A Comprehensive Introduction. India: McGraw Hill Education.

Paper Code: BCA202C

Paper Title: Discrete structure (6 Credits L+T+P = 4+0+2)

The goal of this course is to teach students to Boolean algebra, sets, relations, functions, counting principles, and growth functions so that they may use these concepts in future courses.

On successful completion of the course, students will be able to:

1. Understand the notion of mathematical thinking, mathematical proofs, and algorithmic thinking, and be able to apply them in problem solving.
2. Define mathematical structures (relations, functions, sequences, series, and graphs) and use them to model real life situations
3. Understand asymptotic notation, its significance, and be able to use it to analyse asymptotic performance for some basic algorithmic examples.
4. Understand some basic properties of graphs and related discrete structures, and be able to relate these to practical examples.
5. Understand (trace) and construct simple mathematical proofs using logical arguments.
6. Compare functions and relations with respect to their growth for large values of the input.

Unit	Content	No. of Lectures
1	Introduction: Sets - finite and infinite sets, uncountable infinite sets; functions, relations, properties of binary relations, closure, partial ordering relations; counting - Pigeonhole Principle, permutation and combination; mathematical induction, Principle of Inclusion and Exclusion.	20
2	Growth of Functions: Asymptotic notations, summation formulas and properties, bounding summations, approximation by integrals.	10
3	Recurrence: Recurrence relations, generating functions, linear recurrence relations with constant coefficients and their solution, recursion trees, Master Theorem	10

4	<p>Graph Theory: Basic terminology, models and types, multigraphs and weighted graphs, graph representation, graph isomorphism, connectivity, Euler and Hamiltonian Paths and Circuits, planar graphs, graph coloring, Trees, basic terminology and properties of Trees, introduction to spanning trees.</p>	12								
5	<p>Propositional Logic: Logical connectives, well-formed formulas, tautologies, equivalences, Inference Theory</p>	12								
	<p>Practical:</p> <ol style="list-style-type: none"> 1. Write a Program to create a SETA and determine the cardinality of SET for an input array of elements(repetition allowed) and perform the following operations on the SET: <ol style="list-style-type: none"> a. Is member (a, A): check whether an element belongs to set or not and return value as true/false. b. Power set (A): list all the elements of the power set of A. 2. Create a class SET and take two sets as input from user to perform following SET Operations: <ol style="list-style-type: none"> a. Subset: Check whether one set is a subset of another or not. b. Union and Inter section of two Sets. c. Complement: Assume Universal Set as per the input elements from the user. d. Set Difference and Symmetric Difference between two SETS e. Cartesian Product of Sets. 3. Create a class RELATION, use Matrix notation to represent a relation. Include functions to check if the relation is Reflexive, Symmetric, Anti-symmetric and Transitive. Write a Program to use this class. 4. Use the functions defined in Question 3 to check whether the given relation is: <ol style="list-style-type: none"> a) Equivalent, or b) Partial Order relation, or c) None 5. Write a Program to implement Bubble Sort. Find the number of comparisons during each pass and display the intermediate result. Use the observed values to plot a graph to analyse the complexity of the algorithm. 6. Write a Program to implement Insertion Sort. Find the number of comparisons during each pass and display the intermediate result. Use the observed values to plot a graph to analyse the complexity of the algorithm. 7. Write a Program that generates all the permutations of a given set of digits, with or without repetition. (For example, if the given set is {1,2},the permutations are12 and 21). 8. Write a Program to calculate Permutation and Combination for an input value n and r using the recursive formula of nCr and nPr . 9. For any number n,write a program to list all the solutions of the equation $x_1+x_2+x_3+\dots+x_n= C$, where C is a constant ($C \leq 10$) and $x_1, x_2, x_3, \dots, x_n$ are nonnegative integers using brute force strategy. 10. Write a Program to accept the truth values of variables x and y, and print the truth table of the following logical operations: <table style="width: 100%; border: none;"> <tr> <td>a)Conjunction</td> <td>f)Exclusive NOR</td> </tr> <tr> <td>b)Disjunction</td> <td>g)Negation</td> </tr> <tr> <td>c)ExclusiveOR</td> <td>h) NAND</td> </tr> <tr> <td>d)Conditional</td> <td>i)NOR</td> </tr> </table> 	a)Conjunction	f)Exclusive NOR	b)Disjunction	g)Negation	c)ExclusiveOR	h) NAND	d)Conditional	i)NOR	2 Credits
a)Conjunction	f)Exclusive NOR									
b)Disjunction	g)Negation									
c)ExclusiveOR	h) NAND									
d)Conditional	i)NOR									

	<p>e)Bi-conditional</p> <ol style="list-style-type: none"> 11. Write a Program to store a function(polynomial/exponential), and then evaluate the polynomial.(For example store $f(x)=4n^3+2n+9$ in an array and for a given value of n, say $n=5$,evaluate(i.e. Compute the value of $f(5)$). 12. Write a Program to represent Graphs using the Adjacency Matrices and check if it is a complete graph. 13. Write a Program to accept a directed graph G and compute the in-degree and out-degree of each vertex. 14. Given a graph G, write a Program to find the number of paths of length n between the source and destination entered by the user. 15. Given an adjacency matrix of a graph, write a program to check whether a given set of vertices $\{v_1, v_2, v_3, \dots, v_k\}$ forms an Euler path /Euler Circuit(for circuit assume $v_k=v_1$). 16. Given a full m-ary tree with i internal vertices, Write a Program to find the number of leaf nodes. 	
--	---	--

Reading Books

- Mohapatra, & Liu, C.L. (2012). Elements of Discrete mathematics. 4th edition. McGraw Hill Education.
- Rosen, K.H. (2011). Discrete Mathematics and Its Applications. 7th edition. Tata McGraw Hill Education.
- Albertson, M.O., & Hutchinson, J.P., (1988). Discrete Mathematics with Algorithms. John Wiley and Sons.

Paper Code: BCA301C

Paper Title: Data Structures (6 Credits L+T+P = 4+0+2)

This course seeks to improve student's abilities to solve problems using fundamental data structures such as arrays, stacks, queues, lists, trees, and hash tables. The language C/C++ was selected to comprehend the implementation of these data structures.

On successful completion of the course, students will be able to:

1. To be familiar with fundamental data structures and with the manner in which these data structures can best be implemented; become accustomed to the description of algorithms in both functional and procedural styles
2. To have knowledge of the complexity of basic operations like insert, delete, search on these data structures.
3. Implement and empirically analyze linear and non-linear data structures like Arrays, Stacks, Queues, Lists, Trees, Heaps and Hash tables as abstract data structures.
4. Ability to choose a data structure to suitably model any data used in computer applications.

5. Design programs using various data structures including hash tables, Binary and general search trees, heaps, graphs etc.
6. Implement and know the applications of algorithms for sorting, pattern matching etc.
7. Write programs using recursion for simple problems. Explain the advantages and disadvantages of recursion.

Unit	Content	No. of Lectures
1	Introduction: Definition of data structure, classification of data structure: primitive and non-primitive, linear and nonlinear, ADT, data structure operations. Algorithms : definition, different approaches of designing an algorithm, Complexity, Time Space tradeoff, Asymptotic Notations for Complexity of Algorithms.	5
2	Arrays: Single and Multi-dimensional Arrays, Sparse Matrices (Array and Linked Representation).	6
3	Linked Lists: Singly, Doubly and Circular Lists (Array and Linked representation); Normal and Circular representation of Stack in Lists; Self Organizing Lists; Skip Lists.	8
4	Stacks and Queues: 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. Queue, Representation of Queue, De-queue, Priority Queues.	7
5	Recursion: Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation)	6
6	Trees: Introduction to Tree as a data structure; Binary Trees (Insertion, Deletion, Recursive and Iterative Traversals in Binary Search Trees); Threaded Binary Trees (Insertion, Deletion, Traversals); Height-Balanced Trees (Various operations on AVL Trees).	20
7	Searching and Sorting: Linear Search, Binary Search, Comparison of Linear and Binary Search, Selection Sort, Insertion Sort, Insertion Sort, Shell Sort, Comparison of Sorting Techniques.	6
8	Hashing: 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.	6
	Practical: <ol style="list-style-type: none"> 1. Write a program to search an element from a list. Give users the option to perform Linear or Binary search. Use Template functions. 2. WAP using templates to sort a list of elements. Give users 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 	2 Credits

	<p>linked lists (include a function and also overload operator +).</p> <ol style="list-style-type: none"> 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 a linked list and add two polynomials. 11. WAP to calculate factorial and to compute the factors of a given no. <ol style="list-style-type: none"> (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 numbers. <ol style="list-style-type: none"> (i) with recursion (ii) without recursion 14. WAP to create a Binary Search Tree and include following operations in tree: <ol style="list-style-type: none"> (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 a one-dimensional array. 19. WAP to implement Lower Triangular Matrix using a one-dimensional array. 20. WAP to implement the Upper Triangular Matrix using a one-dimensional array. 21. WAP to implement a Symmetric Matrix using a 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. 23. WAP to implement various operations on AVL Tree. 	
--	--	--

Reading Books

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

Paper Code: BCA302C

Paper Title: Operating Systems (6 Credits, L+T+P = 4+0+2)

Students will learn about several operating systems in this course. Memory management, process management, and file management are a few of the operating system components that are thoroughly discussed.

On successful completion of the course, students will be able to:

1. Describe the important computer system resources and the role of operating system in their management policies and algorithms.
2. To understand various functions, structures and history of operating systems and should be able to specify objectives of modern operating systems and describe how operating systems have evolved over time.
3. Understanding of design issues associated with operating systems.
4. Understand various process management concepts including scheduling, synchronization, and deadlocks.

5. To have a basic knowledge about multithreading.
6. To understand concepts of memory management including virtual memory.
7. To understand issues related to file system interface and implementation, disk management.
8. To understand and identify potential threats to operating systems and the security features designed to guard against them.
9. Implement CPU and disk scheduling algorithms.

Unit	Content	No. of Lectures
1	Introduction: Basic OS functions, resource abstraction, types of operating systems–multiprogramming systems, batch systems, time sharing systems; operating systems for personal computers & workstations, process control & real time systems.	10
2	Operating System Organization: Processor and user modes, kernels, system calls and system programs.	8
3	Process Management: System view of the process and resources, process abstraction, process hierarchy, threads, threading issues, thread libraries; Process Scheduling, non-pre-emptive and preemptive scheduling algorithms; concurrent and processes, critical section, semaphores, methods for inter process communication; deadlocks.	20
4	Memory Management: Physical and virtual address space; memory allocation strategies -fixed and variable partitions, paging, segmentation, virtual memory.	10
5	File and I/O Management: Directory structure, file operations, files allocation methods, device management.	10
6	Protection and Security: Policy mechanism, Authentication, Internal access Authorization.	6
	<p>Practical:</p> <ol style="list-style-type: none"> 1. Write a program (using fork() and/or exec() commands) where parent and child execute: <ol style="list-style-type: none"> a) same program, same code. b) same program, different code. c) before terminating, the parent waits for the child to finish its task. 2. Write a program to report behaviour of Linux kernel including kernel version, CPU type and model. (CPU information) 3. Write a program to report behaviour of Linux kernel including information on configured memory, amount of free and used memory. (memory information) 4. Write a program to print file details including owner access permissions, file access time, where file name is given as argument. 5. Write a program to copy files using system calls. 6. Write a program to implement FCFS scheduling algorithm. 7. Write a program to implement Round Robin scheduling algorithm. 8. Write a program to implement the SJF scheduling algorithm. 9. Write a program to implement a non-preemptive priority-based scheduling 	2 Credits

	<p>algorithm.</p> <p>10. Write a program to implement preemptive priority-based scheduling algorithms.</p> <p>11. Write a program to implement SRJF scheduling algorithm.</p> <p>12. Write a program to calculate the sum of n numbers using a thread library.</p> <p>13. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.</p>	
--	---	--

Reading Books

- A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, John Wiley Publications 2008.
- A.S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.
- G. Nutt, Operating Systems: A Modern Perspective, 2nd Edition Pearson Education 1997.
- W. Stallings, Operating Systems, Internals & Design Principles , 5th Edition, rentice Hall of India. 2008.
- M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.

Paper Code: BCA303C

Paper Title: Computer Networks (6 Credits, L+T+P = 4+0+2)

The principles of data communication and computer networks are covered in this course. It includes study on standard models for layered protocol architecture used to interact between autonomous computers in a network, as well as the key characteristics and challenges of communication protocols at various layers. Introduction to OSI and TCP/IP models are among the topics addressed.

On successful completion of the course, students will be able to:

1. Understand the structure of the Data Communications System and its components. Be familiarize with different network terminologies.
2. Familiarize with contemporary issues in network technologies.
3. Know the layered model approach explained in OSI and TCP/IP network models
4. Identify different types of network devices and their functions within a network.
5. Learn basic routing mechanisms, IP addressing scheme and internetworking concepts.
6. Familiarize with IP and TCP Internet protocols.
7. To understand major concepts involved in the design of WAN, LAN and wireless networks.
8. Learn basics of network configuration and maintenance.
9. Know the fundamentals of network security issues.

10. Describe, analyze and compare different data link, network, and transport layer protocols.
11. Design/implement data link and network layer protocols in a simulated networking environment.

Unit	Content	No. of Lectures
1	Introduction to Computer Networks: Network definition; network topologies; network classifications; network protocol; layered network architecture; overview of OSI reference model; overview of TCP/IP protocol suite.	8
2	Data Communication Fundamentals and Techniques: Analog and digital signal; data-rate limits; digital to digital line encoding schemes; pulse code modulation; parallel and serial transmission; digital to analog modulation-; multiplexing techniques-FDM,TDM; transmission media.	10
3	Networks Switching Techniques and Access Mechanisms: Circuit switching; packet switching-connectionless datagram switching, connection-oriented virtual circuit switching; dial-up modems; digital subscriber line;cable TV for data transfer.	10
4	Data Link Layer Functions and Protocol: Error detection and error correction techniques; data-link control- framing and flow control; error recovery protocols-stop and wait ARQ, go-back-nARQ; Point to Point Protocol on the Internet.	10
5	Multiple Access Protocol and Networks: CSMA/CD protocols; Ethernet LANS; connecting LAN and back-bone networks- repeaters, hubs, switches, bridges, router and gateways.	5
6	Networks Layer Functions and Protocols: Routing; routing algorithms; network layer protocol of Internet-IP protocol, Internet control protocols.	6
7	Transport Layer Functions and Protocols: Transport services-error and flow control, Connection establishment and release-three way handshake;	6
8	Overview of Application Layer Protocol: Overview of DNS protocol; overview of WWW & HTTP protocol.	5
	Practical: <ol style="list-style-type: none"> 1. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel. 2. Simulate and implement a stop and wait protocol for noisy channels. 3. Simulate and implement go back to sliding window protocol. 4. Simulate and implement selective repeat sliding window protocol. 5. Simulate and implement distance vector routing algorithm 6. Simulate and implement Dijkstra algorithm for shortest path routing. 	2 Credits

Reading Books

- B.A.Forouzan: Data Communications and Networking, Fourth edition, THM,2007.
- A.S.Tanenbaum: Computer Networks,Fourth edition,PHI,2002.

Paper Code: BCA401C

Paper Title: Design and Analysis of Algorithms (6 Credits, L+T+P = 4+0+2)

The goal of this course is to teach students how to design and analyse algorithms for efficiency and correctness. For efficient algorithm design, the course emphasizes the differences between various problem-solving techniques.

On successful completion of the course, students will be able to:

1. To learn good principles of algorithm design;
2. To learn how to analyse algorithms and estimate their worst-case and average case behavior (in easy cases);
3. Write algorithms choosing the best one or a combination of two or more of the algorithm design techniques: Iterative, divide-n-conquer, Greedy, Dynamic Programming using appropriate data structures.
4. To learn how to apply their theoretical knowledge in practice (via the practical component of the course).
5. Re-write a given algorithm replacing the (algorithm design) technique used with a more appropriate/efficient (algorithm design) technique.

Unit	Content	No. of Lectures
1	Introduction Basic Design and Analysis techniques of Algorithms, Correctness of Algorithm.	7
2	Algorithm Design Techniques Iterative techniques, Divide and Conquer, Dynamic Programming, Greedy Algorithms.	10
3	Sorting and Searching Techniques Elementary sorting techniques–Bubble Sort, Insertion Sort, Merge Sort, Advanced Sorting techniques-Heap Sort, Quick Sort, Sorting in Linear Time-Bucket Sort, Radix Sort and Count Sort, Searching Techniques, Medians & Order Statistics, complexity analysis;	15
4	Algorithm Design Techniques Decision Trees	5
5	Balanced Trees Red-black Trees	5
6	Advanced Analysis Technique Amortized analysis	5
7	Graphs Graph Algorithms–Breadth First Search, Depth First Search and its Applications, Minimum Spanning Trees	8
8	String Processing String matching, KMP technique.	5
	Practical: 1. (i)ImplementInsertionSort(The program should report the number of	2 Credits

	comparisons) (ii) ImplementMergeSort(The program should report the number of comparisons) 2. ImplementHeapSort(The program should report the number of comparisons) 3. ImplementRandomizedQuicksort(The program should report the number of comparisons) 4. ImplementRadix Sort 5. Create a Red-BlackTree and perform following operations on it: a. (i)Insert anode b. (ii)Delete anode c. (iii)Search for a number & also report the color of the node containing this number. 6. Writeaprogramtodeterminethe LCSof twogivensequences 7. ImplementBreadth-First Search in a graph 8. ImplementDepth-FirstSearchin a graph 9. Write a program to determine the minimum spanning tree of a graph	
--	--	--

Reading Books

- T.H.Cormen,CharlesE. Leiserson, Ronald L.Rivest,CliffordStein Introduction to Algorithms,PHI, 3rd Edition 2009
- Sarabasse & A.V. Gelder Computer Algorithm – Introduction to Design and Analysis,Publisher– Pearson 3rd Edition 1999

Paper Code: BCA402C

Paper Title: Software Engineering (6 Credits, L+T+P = 4+0+2)

Fundamental Software Engineering methodologies and strategies for software development are covered in this course. In addition, the students create a case study using the relevant software model.

On successful completion of the course, students will be able to:

1. Analyze and model customer’s requirements and model its software design.
2. Ability to apply software engineering principles and techniques.
3. Use a suitable software model for the problem at hand.
4. Estimate cost and efforts required in building software.
5. Analyze and compute impact of various risks involved in software development.
6. Design and build test cases, and to perform software testing.

Unit	Content	No. of Lectures
1	Introduction 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).	10

2	Requirement Analysis Software Requirement Analysis, Initiating Requirement Engineering Process, Requirement Analysis and Modeling Techniques, Flow Oriented Modeling, Need for SRS, Characteristics and Components of SRS.	8
3	Software Project Management Estimation in Project Planning Process, Project Scheduling	7
4	Risk Management Software Risks, Risk Identification, Risk Projection and Risk Refinement, RMMM Plan.	7
5	Quality Management Quality Concepts, Software Quality Assurance, Software Reviews, Metrics for Process and Projects	6
6	Design Engineering 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.	12
7	Testing Strategies & Tactics Software Testing Fundamentals, Strategic Approach to Software Testing, Test Strategies for Conventional Software, Validation Testing, System testing, Black-Box Testing, White-Box Testing and their type, Basis Path Testing.	10
Practical:		2 Credits
Sl. No	Practical title	
1	Software Life Cycle Model: <ul style="list-style-type: none"> ● Understanding of Process model 	
2	Requirement analysis: <ul style="list-style-type: none"> ● Creating data flow diagram ● Data dictionary, use cases 	
3	Project Management: <ul style="list-style-type: none"> ● Computing FP ● Effort ● Schedule, Risk Table, Time line chart 	
4	Design: <ul style="list-style-type: none"> ● Architectural Design ● Data Design, Component Level Design 	
5	Testing: <ul style="list-style-type: none"> ● Basis Path Testing 	

Reading Books

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

Paper Code: BCA403C

Paper Title: Database Management System (6 Credits, L+T+P = 4+0+2)

The importance of a database, relational data model, schema design and normalization, transaction processing, indexing, and the related data structures (files and B+-trees) are covered in this course.

On successful completion of the course, students will be able to:

1. Describe major components of DBMS and their functions
2. Model an application's data requirements using conceptual modeling tools like ER diagrams and design database schemas based on the conceptual model.
3. Write queries in relational algebra as well as in SQL
4. Normalize a given database schema to avoid data anomalies and data redundancy.
5. Describe the notions of indexes, views, constraints and transactions.

Unit	Content	No. of Lectures
1	Introduction Characteristics of database approach, data models, database system architecture and data independence	6
2	Entity Relationship(ER) Modeling Entity types, relationships, constraints.	8
3	Relation data model Relational model concepts, relational constraints, relational algebra, SQL	20
4	Database design Mapping ER/EER model to relational database, functional dependencies, Lossless decomposition and Normalization (upto BCNF)	15

5	Transaction Processing ACID properties, concurrency control	3																																																																	
6	File Structure and Indexing Operations on files, File of Unordered and ordered records, overview of File organizations, Indexing structures for files (Primary index, secondary index, clustering index), Multilevel indexing using B and B+ trees.	8																																																																	
<p>Practical: Create and use the following database schema to answer the given queries. EMPLOYEE Schema</p> <table border="1" data-bbox="302 554 1341 1138"> <thead> <tr> <th>FIELD</th> <th>TYPE</th> <th>NULL</th> <th>KEY</th> <th>DEFAULT</th> </tr> </thead> <tbody> <tr> <td>Eno</td> <td>Char(3)</td> <td>NO</td> <td>PRI</td> <td>NIL</td> </tr> <tr> <td>Ename</td> <td>Varchar(50)</td> <td>NO</td> <td></td> <td>NIL</td> </tr> <tr> <td>Job_type</td> <td>Varchar(50)</td> <td>NO</td> <td></td> <td>NIL</td> </tr> <tr> <td>Manager</td> <td>Char(3)</td> <td>Yes</td> <td>FK</td> <td>NIL</td> </tr> <tr> <td>Hire_date</td> <td>Date</td> <td>NO</td> <td></td> <td>NIL</td> </tr> <tr> <td>Dno</td> <td>Integer</td> <td>YES</td> <td>FK</td> <td>NIL</td> </tr> <tr> <td>Commission</td> <td>Decimal(10,2)</td> <td>YES</td> <td></td> <td>NIL</td> </tr> <tr> <td>Salary</td> <td>Decimal(7,2)</td> <td>NO</td> <td></td> <td>NIL</td> </tr> </tbody> </table> <p>DEPARTMENT Schema</p> <table border="1" data-bbox="302 1243 1341 1503"> <thead> <tr> <th>FIELD</th> <th>TYPE</th> <th>NULL</th> <th>KEY</th> <th>DEFAULT</th> </tr> </thead> <tbody> <tr> <td>Dno</td> <td>Integer</td> <td>No</td> <td>PRI</td> <td>NIL</td> </tr> <tr> <td>Dname</td> <td>Varchar(50)</td> <td>Yes</td> <td></td> <td>NIL</td> </tr> <tr> <td>Location</td> <td>Varchar(50)</td> <td>Yes</td> <td></td> <td>New Delhi</td> </tr> </tbody> </table> <p>Query List:</p> <ol style="list-style-type: none"> Query to display Employee Name, Job, Hire Date, Employee Number; for each employee with the Employee Number appearing first. Query to display unique Jobs from the Employee Table. Query to display the Employee Name concatenated by a Job separated by a comma. Query to display all the data from the Employee Table. Separate each Column by a comma and name the said column as THE_OUTPUT. 		FIELD	TYPE	NULL	KEY	DEFAULT	Eno	Char(3)	NO	PRI	NIL	Ename	Varchar(50)	NO		NIL	Job_type	Varchar(50)	NO		NIL	Manager	Char(3)	Yes	FK	NIL	Hire_date	Date	NO		NIL	Dno	Integer	YES	FK	NIL	Commission	Decimal(10,2)	YES		NIL	Salary	Decimal(7,2)	NO		NIL	FIELD	TYPE	NULL	KEY	DEFAULT	Dno	Integer	No	PRI	NIL	Dname	Varchar(50)	Yes		NIL	Location	Varchar(50)	Yes		New Delhi	2 Credits
FIELD	TYPE	NULL	KEY	DEFAULT																																																															
Eno	Char(3)	NO	PRI	NIL																																																															
Ename	Varchar(50)	NO		NIL																																																															
Job_type	Varchar(50)	NO		NIL																																																															
Manager	Char(3)	Yes	FK	NIL																																																															
Hire_date	Date	NO		NIL																																																															
Dno	Integer	YES	FK	NIL																																																															
Commission	Decimal(10,2)	YES		NIL																																																															
Salary	Decimal(7,2)	NO		NIL																																																															
FIELD	TYPE	NULL	KEY	DEFAULT																																																															
Dno	Integer	No	PRI	NIL																																																															
Dname	Varchar(50)	Yes		NIL																																																															
Location	Varchar(50)	Yes		New Delhi																																																															

5. Query to display the Employee Name and Salary of all the employees earning more than \$2850.
6. Query to display Employee Name and Department Number for the Employee No= 7900.
7. Query to display Employee Name and Salary for all employees whose salary is not in the range of \$1500 and \$2850.
8. Query to display Employee Name and Department No. of all the employees in Dept 10 and Dept 30 in the alphabetical order by name.
9. Query to display Name and Hire Date of every Employee who was hired in 1981.
10. Query to display Name and Job of all employees who don't have a current Manager.
11. Query to display the Name, Salary and Commission for all the employees who earn commission.
12. Sort the data in descending order of Salary and Commission.
13. Query to display the name of all the employees where the third letter of their name is 'A'.
14. Query to display the name of all employees either have two 'R's or have two 'A's in their name and are either in Dept No = 30 or their Manager's Employee No = 7788.
15. Query to display Name, Salary and Commission for all employees whose Commission Amount is 14 greater than their Salary increased by 5%.
16. Query to display the name of the most recent joining employee.
17. Query to display Name, Hire Date and Salary Review Date which is the 1st Monday after six months of employment.
18. Query to display Name and calculate the number of months between today and the date each employee was hired.
19. Query to display the following for each employee <E-Name> earns < Salary> monthly but wants < 3 * Current Salary >. Label the Column as Dream Salary.
20. Query to display Name with the 1st letter capitalized and all other lower case and length of their name of all the employees whose name starts with 'J', 'A' and 'M'.
21. Query to display Name, Hire Date and Day of the week on which the employee started.
22. Query to display Name, Department Name and Department No for all the employees.
23. Query to display Unique Listing of all Jobs that are in Department # 30.
24. Query to display Name, Dept Name of all employees who have an 'A' in their name.
25. Query to display Name, Job, Department No. And Department Name for all the employees working at the Dallas location.
26. Query to display Name and Employee no. Along with their Manager's Name and the Manager's employee no; along with the Employees' Name who do not have a Manager.
27. Query to display Name, Dept No. And Salary of any employee whose department No. and salary matches both the department no. And the salary of any employee who earns a commission.
28. Query to display Name and Salaries represented by asterisks, where each

	<p>asterisk (*) signifies \$100.</p> <p>29. Query to display the Highest, Lowest, Sum and Average Salaries of all the employees</p> <p>30. Query to display the number of employees performing the same Job type functions.</p> <p>31. Query to display the no. of managers without listing their names.</p> <p>32. Query to display the Department Name, Location Name, No. of Employees and the average salary for all employees in that department.</p> <p>33. Query to display Name and Hire Date for all employees in the same dept. as Blake.</p> <p>34. Query to display the Employee No. And Name for all employees who earn more than the average salary.</p> <p>35. Query to display Employee Number and Name for all employees who work in a department with any employee whose name contains a 'T'.</p> <p>36. Query to display the names and salaries of all employees who report to King.</p> <p>37. Query to display the department no, name and job for all employees in the Sales department.</p>	
--	--	--

Reading Books

- R. Elmasri, S.B. Navathe, Fundamentals of Database Systems 6th Edition, Pearson Education, 2010.
- R. Ramakrishanan, J. Gehrke, Database Management Systems 3rd Edition, McGraw-Hill, 2002
- A. Silberschatz, H.F. Korth, S. Sudarshan, Database System Concepts 6th Edition, McGraw Hill, 2010.
- R. Elmasri, S.B. Navathe Database Systems Models, Languages, Design and application Programming, 6th Edition, Pearson Education, 2013.

Paper Code: BCA501C

Paper Title: Internet Technologies (6 Credits, L+T+P = 4+0+2)

This course covers the event handlers which make web pages dynamic in nature with more features. Students will get an understanding of how to design a search engine with advanced functionalities.

On successful completion of the course, students will be able to:

1. To understand the use of Objects in Java
2. Use JavaScript technologies to make a website highly responsive, more efficient and user friendly
3. Demonstrate an understanding of and the ability to use the database using JDBC

4. Understand the functionality of JSP and Java Beans.
5. Develop an ability to create basic Web pages with HTML.
6. Implement a simple search engine.

Unit	Content	No. of Lectures
1	Java Use of Objects, Array and Array List class	5
2	Java Script Data types, operators, functions, control structures, events and event handling.	15
3	JDBC JDBC Fundamentals, Establishing Connectivity and working with connection interface, working with statements, Creating and Executing SQL Statements, Working with Result Set Objects.	10
4	JSP Introduction to Java Server 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.	20
5	Java Beans Java Beans Fundamentals, JAR files, Introspection, Developing a simple Bean, Connecting to DB	10
	Practical: Create event driven program for following: <ol style="list-style-type: none"> 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. 4. Enter a list of positive numbers terminated by Zero. Find the sum and average of these numbers. 5. A person deposits Rs1000 in a fixed account yielding 5 % interest. Compute the amount in the account at the end of each year for n years. 6. Read n numbers. Count the number of negative numbers, positive numbers and zeros in the list 	2 Credits

Reading Books

- Ivan Bayross, Web Enabled Commercial Application Development Using Html, Dhtml, java script, Perl Cgi , BPB Publications, 2009.
- CayHorstmann, BIG Java, WileyPublication , 3rd Edition., 2009
- HerbertSchildt, Java7,TheCompleteReference, 8thEdition, 2009.
- JimKeogh,TheCompleteReferenceJ2EE, TMH,2002.
- O'Reilly,JavaServerPages,Hans Bergsten,Third Edition,2003.

Paper Code: BCA502C

Paper Title: Theory of Computation (6 Credits, L+T+P = 5+1+0)

This course introduces the finite automaton, pushdown automaton, and Turing machine as formal models of computation, as well as their relationships with formal languages. Students will also learn about computer machines' limits.

On successful completion of the course, students will be able to:

1. Design a finite automaton, pushdown automaton or a Turing machine for a problem at hand.
2. Apply pumping lemma to prove that a language is non-regular/non-context-free.
3. Describe limitations of a computing machine.

Unit	Content	No. of Lectures
1	Languages Alphabets, string, language, Basic Operations on language, Concatenation, KleeneStar	5
2	Finite Automata and Regular Languages Regular Expressions, Transition Graphs, Deterministic and non-deterministic finite automata, NFA to DFA Conversion, Regular languages and their relationship with finite automata, Pumping lemma and closure properties of regular languages.	15
3	Context free languages Context free grammars, parse trees, ambiguities in grammars and languages, Pushdown automata (Deterministic and Non-deterministic), Pumping Lemma, Properties of context free languages, normal forms.	10
4	Turing Machines and Models of Computations RAM, Turing Machine as a model of computation, Universal Turing Machine, Language acceptability, decidability, halting problem, Recursively enumerable and recursive languages, unsolvability problems.	20

Reading Books

- Daniel I.A.Cohen, Introduction to computer theory, John Wiley, 1996
- Lewis & Papadimitriou, Elements of the theory of the theory of computation, PHI 1997.
- Hopcroft, Aho, Ullman, Introduction to Automata theory, Language & Computation – 3rd Edition, Pearson Education. 2006
- P.Linz, An Introduction to Formal Language and Automata 4th edition Publication Jones Bartlett, 2006

Paper Code: BCA601C

Paper Title: Artificial Intelligence (6 Credits, L+T+P = 4+0+2)

The fundamental principles and techniques of Artificial Intelligence are covered in this course (AI). The purpose of this course is to introduce intelligent agents and reasoning, as well as heuristic search techniques, game play, knowledge representation, and reasoning with uncertain knowledge.

On successful completion of the course, students will be able to:

1. Explain what constitutes "Artificial" Intelligence and how to identify systems with Artificial Intelligence.
2. Identify problems that are amenable to solution by specific AI methods
3. Identify appropriate AI technique for the problem at hand (e.g., as a search problem, as a constraint satisfaction problem, as a planning problem, etc).
4. Implement basic AI algorithms (e.g., standard search or constraint propagation algorithms).
5. Design and perform an empirical evaluation of different algorithms on problem formalization, and state the conclusions that the evaluation supports.
6. Explain the limitations of current Artificial Intelligence techniques.

Unit	Content	No. of Lectures
1	Introduction Introduction to Artificial Intelligence, Background and Applications, Turing Test and Rational Agent approaches to AI, Introduction to Intelligent Agents, the irstructure, behaviour and environment.	5
2	Problem Solving and Searching Techniques Problem Characteristics, Production Systems, Control Strategies, Breadth First Search, Depth First Search, Hill climbing and its Variations, Heuristics Search Techniques: Best First Search, A*algorithm, Constraint Satisfaction Problem, Means-End Analysis, Introduction to Game Playing, Min-Max and Alpha-Beta pruning algorithms.	15
3	Knowledge Representation Introduction to First Order Predicate Logic, Resolution Principle, Unification, Semantic Nets, Conceptual Dependencies, Frames, and Scripts, Production Rules, Conceptual Graphs. Programming in Logic (PROLOG)	10
4	Dealing with Uncertainty and Inconsistencies Truth Maintenance System, Default Reasoning, Probabilistic Reasoning, Bayesian Probabilistic Inference, Possible World Representations.	20
5	Understanding Natural Languages Parsing Techniques, Context-Free and Transformational Grammars, Recursive and Augmented, Transition Nets.	
	Practical: <ol style="list-style-type: none"> 1. Write a prolog program to calculate the sum of two numbers. 2. Write a prolog program to find the maximum of two numbers. 3. Write a prolog program to calculate the factorial of a given number. 4. Write a prolog program to calculate the nth Fibonacci number. 	2 Credits

	<ol style="list-style-type: none"> 5. Write a prolog program, insert_nth(item, n, into_list, result) that asserts that result is the list into_list with item inserted as the n'th element into every list at all levels. 6. Write a Prolog program to remove the Nth item from a list. 7. Write a Prolog program, remove_nth(Before, After) that asserts the After list is the Before list with the removal of every n'th item from every list at all levels. 8. Write a Prolog program to implement append for two lists. 9. Write a Prolog program to implement palindrome (List). 10. Write a Prolog program to implement max(X,Y,Max) so that Max is the greater of two numbers X and Y. 11. Write a Prolog program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List. 12. Write a Prolog program to implement sumlist(List,Sum) so that Sum is the sum of a given list of numbers List. 13. Write a Prolog program to implement two predicates evenlength(List) and odd length(List) so that they are true if their argument is a list of even or odd length respectively. 14. Write a Prolog program to implement reverse (List,ReversedList) that reverses lists. 15. Write a Prolog program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List using cut predicate. 16. Write a Prolog program to implement GCD of two numbers. 17. Write a Prolog program that implements Semantic Networks/Frame Structures. 18. Write a Prolog program to implement max(X,Y,Max) so that Max is the greater of two numbers X and Y. 19. Write a Prolog program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List. 20. Write a Prolog program to implement sumlist(List,Sum) so that Sum is the sum of a given list of numbers List. 21. Write a Prolog program to implement two predicates evenlength (List) and odd length (List) so that they are true if their argument is a list of even or odd length respectively. 22. Write a Prolog program to implement reverse (List,ReversedList) that reverses lists. 23. Write a Prolog program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List using cut predicate. 24. Write a Prolog program to implement GCD of two numbers. 25. Write a prolog program that implements Semantic Networks/Frame Structures. 	
--	--	--

Reading Books

- DAN.W. Patterson, Introduction to AI and Expert Systems – PHI, 2007.
- Russell & Norvig, Artificial Intelligence-A Modern Approach, LPE, Pearson Prentice Hall, 2nd edition, 2005.
- Rich & Knight, Artificial Intelligence – Tata McGraw Hill, 2nd edition, 1991.
- W.F. Clocksin and Mellish, Programming in PROLOG, Narosa Publishing House, 3rd edition, 2001.
- Ivan Bratko, Prolog Programming for Artificial Intelligence, Addison-Wesley, Pearson Education, 3rd edition, 2000.

Paper Code: BCA602C

Paper Title: Computer Graphics (6 Credits L+T+P = 4+0+2)

This course introduces the fundamental concepts of computer graphics, with an emphasis on modeling, rendering, and interaction. The focus of the course is on the fundamental ideas required to create, operate, and comprehend computer graphics systems.

On successful completion of the course, students will be able to:

1. Acquire familiarity with the concepts and relevant mathematics of computer graphics.
2. Ability to implement various algorithms to scan, convert the basic geometrical primitives, transformations, area filling, clipping.
3. Describe the importance of viewing and projections.
4. Design basic graphics application programs.
5. Design applications that display graphic images to given specifications.
6. Implement visible surface determination algorithms, Illumination models and surface rendering methods, color models

Unit	Content	No. of Lectures
1	Introduction Basic elements of Computer graphics, Applications of Computer Graphics.	5
2	Graphics Hardware Architecture of Raster and Random scan display devices, input/output devices.	8
3	Fundamental Techniques in Graphics Raster scan line, circle and ellipse drawing, thick primitives, Polygon filling, line and polygon clipping algorithms, 2D and 3D Geometric Transformations, 2D and 3D Viewing Transformations (Projections- Parallel and Perspective), Vanishing points.	22
4	Geometric Modeling Representing curves & Surfaces.	10
5	Visible Surface determination Hidden surface elimination.	8
6	Surface Rendering Illumination and shading models. Basic color models and Computer Animation.	7
	Practical: 1. Write a program to implement Bresenham's line drawing algorithm. 2. Write a program to implement the midpoint circle drawing algorithm. 3. Write a program to clip a line using Cohen and Sutherland line clipping algorithm. 4. Write a program to clip a polygon using Sutherland Hodgeman algorithm. 5. Write a program to apply various 2D transformations on a 2D	2 Credits

	object (use homogeneous coordinates). 6. Write a program to apply various 3D transformations on a 3D object and then apply parallel and perspective projection on it. 7. Write a program to draw the Hermite/Bezier curve.	
--	--	--

Reading Books

- J.D.Foley, A. Van Dam, Feiner, Hughes Computer Graphics Principles & Practice 2nd edition, Publication Addison Wesley 1990.
- D.Hearn, Baker: Computer Graphics, Prentice Hall of India 2008.
- D.F.Rogers, Procedural Elements for Computer Graphics, McGraw Hill 1997.
- D.F.Rogers, Adams Mathematical Elements for Computer Graphics, McGraw Hill 2nd edition 1989.

Discipline Specific Elective

Paper Code: DSE01

Paper Title: Computer Fundamentals (6 Credits, L+T+P = 4+0+2)

This course is aimed to provide an overview of computer systems and its functionality. It discusses the different components of computer systems and their organization to accomplish various tasks assigned to computers. This course also covers the concepts of number systems and its representation in computers.

On successful completion of the course, students will be able to:

1. Understand the basics of the computing environment, how computers work and its types.

2. Understand the representation of numbers, alphabets and other characters in a computer system.
3. Learn about the different components of computer systems, which helps computers to work.
4. Understand the basic architecture and organization of computer system
5. Get an idea of emerging technologies and their applications to solve real world problems

Unit	Content	No. of Lectures
1	Introduction: Introduction to computer system, uses, types.	6
2	Data Representation: Number systems and character representation.	12
3	Human Computer Interface: Types of software, Operating system as user interface, utility programs.	6
4	Devices: Input and output devices (with connections and practical demo), keyboard, mouse, joystick, scanner, OCR, OMR, barcode reader, web camera, monitor, printer, plotter.	10
5	Memory: Primary, secondary, auxiliary memory, RAM, ROM, cache memory, hard disks, optical disks.	6
6	Computer Organisation and Architecture: C.P.U., registers, system bus, main memory unit, cache memory, Inside a computer, SMPS, Motherboard, Ports and Interfaces, expansion cards, ribbon cables, memory chips, processors.	12
7	Overview of Emerging Technologies: Bluetooth, cloud computing, big data, data mining, mobile computing and embedded systems.	8
	<p>Practical: Practical exercises based on MS Office/ Open Office tools using document preparation and spreadsheet handling packages.</p> <p>MS Word</p> <ol style="list-style-type: none"> 1. Prepare a grocery list having four columns (Serial number, The name of the product, quantity and price) for the month of April, 06. <ul style="list-style-type: none"> • Font specifications for Title (Grocery List): 14-point Arial font in bold and italics. • The headings of the columns should be in 12-point and bold. • The rest of the document should be in 10-point Times New Roman. 	2 Credits

- Leave a gap of 12-points after the title.
2. Create a **telephone directory**.
 - The heading should be 16-point Arial Font in bold
 - The rest of the document should use 10-point font size
 - Other headings should use 10-point Courier New Font.
 - The footer should show the page number as well as the date last updated.
 3. Design a **time-table form** for your college.
 - The first line should mention the name of the college in 16-point Arial Font and should be bold.
 - The second line should give the course name/teacher's name and the department in 14-point Arial.
 - Leave a gap of 12-points.
 - The rest of the document should use 10-point Times New Roman font.
 - The footer should contain your specifications as the designer and date of creation.
 4. BPB Publications plans to release a new book designed as per your syllabus. Design the **first page of the book** as per the given specifications.
 - The title of the book should appear in bold using a 20-point Arial font.
 - The name of the author and his qualifications should be in the center of the page in 16-point Arial font.
 - At the bottom of the document should be the name of the publisher and address in 16-point Times New Roman.
 - The details of the offices of the publisher (only location) should appear in the footer.
 5. Create the following one page documents.
 1. Compose a note inviting friends to a get-together at your house, Including a list of things to bring with them.
 2. Design a certificate in landscape orientation with a border around the document.
 3. Design a Garage Sale sign.
 4. Make a sign outlining your rules for your bedroom at home, using a numbered list.

6. Create the following documents:

- a) A newsletter with a headline and 2 columns in portrait orientation, including at least one image surrounded by text.
- b) Use a newsletter format to promote upcoming projects or events in your classroom or college.

7. Convert following text to a table, using comma as delimiter

Type the following as shown (do not bold).

Color, Style, Item

Blue, A980, Van

Red, X023, Car

Green, YL724, Truck

Name, Age, Sex

Bob, 23, M

Linda, 46, F

Tom, 29, M

8. Enter the following data into a table given on the next page.

Salesperson	Dolls	Trucks	Puzzles
Kennedy, Sally	1327	1423	1193
White, Pete	1421	3863	2934
Pillar, James	5214	3247	5467
York, George	2190	1278	1928
Banks, Jennifer	1201	2528	1203
Atwater, Kelly	4098	3079	2067

Pillar, James	5214	3247	5467
York, George	2190	1278	1928
Banks, Jennifer	1201	2528	1203
Atwater, Kelly	4098	3079	2067

Add a column Region (values: S, N, N,S,S,S) between the Salesperson and Dolls columns to the given table Sort your table data by Region and within Region by Salesperson in ascending order:

In this exercise, you will add a new row to your table, place the word "Total" at the bottom of the Salesperson column, and sum the Dolls, Trucks, and Puzzles columns.

9. Wrapping of text around the image.

10. Following features of menu option must be covered

FILE	Complete menu
EDIT	Complete menu
VIEW	Complete menu
INSERT	Complete menu
FORMAT	Complete menu
TABLE	Complete menu
WINDOW	Complete menu
HELP	Complete menu
TOOLS	All options except Online collaboration, Tools on Macro, Templates

MS Excel

1. Enter the Following data in Excel Sheet

REGIONAL SALES PROJECTION

State	Qtr1	Qtr2	Qtr3	QTR4	Qtr Total	Rate Amount
Delhi	2020	2400	2100	3000	15	
Punjab	1100	1300	1500	1400	20	
U.P.	3000	3200	2600	2800	17	
Haryana	1800	2000	2200	2700	15	
Rajasthan	2100	2000	1800	2200	20	

TOTAL

AVERAGE

(a) Apply Formatting as follow:

- i. Title in TIMES NEW ROMAN
- ii. Font Size - 14
- iii. Remaining text - ARIAL, Font Size -10
- iv. State names and Qtr. Heading Bold, Italic with Gray Fill Color.
- v. Numbers in two decimal places.
- vi. Qtr. Heading in center Alignment.
- vii. Apply Border to whole data.

(b) Calculate State and Qtr. Total

(c) Calculate Average for each quarter

(d) Calculate Amount = Rate * Total.

2. Given the following worksheet

	A	B	C	D
1	Roll No.	Name	Marks	Grade
2	1001	Sachin	99	
3	1002	Sehwag	65	
4	1003	Rahul	41	
5	1004	Sourav	89	
6	1005	Har Bhajan	56	

Calculate the grade of these students on the basis of following guidelines:

If Marks	Then Grade
≥ 80	A+
$\geq 60 < 80$	A
$\geq 50 < 60$	B
< 50	F

3. Given the following worksheet

	A	B	C	D	E	F	G
1	Salesman						
2	No.	Qtr1	Qtr2	Qtr3	Qtr4	Total	Commission
3	S001	5000	8500	12000	9000		
4	S002	7000	4000	7500	11000		
5	S003	4000	9000	6500	8200		
6	S004	5500	6900	4500	10500		

7	S005	7400	8500	9200	8300		
8	S006	5300	7600	9800	6100		

Calculate the commission earned by the salesmen on the basis of following Candidates:

If Total Sales	Commission
< 20000	0% of sales
> 20000 and < 25000	4% of sales
> 25000 and < 30000	5.5% of sales
> 30000 and < 35000	8% of sales
>= 35000	11% of sales

The total sales is the sum of sales of all the four quarters.

4. A company XYZ Ltd. pays a monthly salary to its employees which consists of basic salary, allowances & deductions. The details of allowances and deductions are as follows:

- HRA Dependent on Basic

30% of Basic if Basic \leq 1000

25% of Basic if Basic $>$ 1000 & Basic \leq 3000

20% of Basic if Basic $>$ 3000

- DA Fixed for all employees, 30% of Basic

- Conveyance Allowance

Rs. 50/- if Basic is \leq 1000

Rs. 75/- if Basic $>$ 1000 & Basic \leq 2000

Rs. 100 if Basic $>$ 2000

- Entertainment Allowance

NIL if Basic is \leq 1000

Rs. 100/- if Basic $>$ 1000

Deductions

- Provident Fund 6% of Basic

- Group Insurance Premium Rs. 40/- if Basic is \leq 1500

Rs. 60/- if Basic $>$ 1500 & Basic \leq 3000

Rs. 80/- if Basic $>$ 3000

Calculate the following.

Gross Salary = Basic + HRA + DA + Conveyance + Entertainment

Total deduction = Provident Fund + Group Insurance Premium

Net Salary = Gross Salary – Total Deduction

5. Create Payment Table for a fixed Principal amount, variable rate of interests and time in the format below:

No. of Instalments	5%	6%	7%	8%	9%
3	xx	xx	xx	xx	xx
4	xx	xx	xx	xx	xx
5	xx	xx	xx	xx	xx
6	xx	xx	xx	xx	xx

6. Use an array formula to calculate Simple Interest for given principal amounts given the rate of Interest and time

Rate of Interest	8%
Time	5 Years
Principal	Simple Interest
1000	?
18000	?
5200	?

7. The following table gives a year wise sale figure of five salesmen in Rs.

Salesman	2000	2001	2002	2003
S1	10000	12000	20000	50000
S2	15000	18000	50000	60000
S3	20000	22000	70000	70000
S4	30000	30000	100000	80000
S5	40000	45000	125000	90000

b) Calculate the commission to be paid by salesman under the condition.

i) If total sales >4,00,000 give 5% commission on total sale made by the salesman.

(ii) Otherwise give 2% commission.

(e) Draw a bar graph representing the sale made by each salesman.

(f) Draw a pie graph representing the sale made by a salesman in 2000.

8. Enter the following data in Excel Sheet

PERSONAL BUDGET FOR FIRST

QUARTER Monthly Income (Net): 1,475

Expenses	January	February	March	Quarter Total	Quarter Average
Rent	600.00	600.00	600.00		
Telephone	48.25	43.50	60.00		
Utilities	67.27	110.00	70.00		
Credit Card	200.00	110.00	70.00		
Oil	100.00	150.00	90.00		
AV to Insurance	150.00				
Cable TV	40.75	40.75	40.75		
Monthly Total					

Calculate Quarter total and Quarter average.

(a) Calculate Monthly total.

- (b) Surplus = Monthly income - Monthly total.
- (c) What would be the total surplus if monthly income is 1500.
- (d) How much does telephone expense for March differ from quarter average.
- (e) Create a 3D column graph for telephone and utilities.
- (f) Create a pie chart for monthly expenses.
9. Enter the following data in Excel Sheet

	TOTAL REVENUE EARNED FOR SAM'S BOOKSTALL				
Publisher Name	1997	1998	1999	2000	Total
A	Rs. 1000.00	Rs. 1100.00	Rs. 1300.00	Rs. 800.00	
B	Rs. 1500.00	Rs. 700.00	Rs. 1000.00	Rs. 2000.00	
C	Rs. 700.00	Rs. 900.00	Rs. 1500.00	Rs. 600.00	
D	Rs. 1200.00	Rs. 500.00	Rs. 200.00	Rs. 1100.00	
E	Rs. 800.00	Rs. 1000.00	Rs. 3000.00	Rs. 560.00	

- a. Compute the total revenue earned.
- b. Plot the line chart to compare the revenue of all publishers for 4 years.
- c. Chart Title should be 'Total Revenue of sam's Bookstall (1997-2000)'
- d. Give appropriate categories and value axis titles.
10. Generate 25 random numbers between 0 & 100 and find their sum, average and count. How many no. are in range 50-60.

Reading list:

- A. Goel, Computer Fundamentals, Pearson Education, 2010.
- P. Aksoy, L. DeNardis, Introduction to Information Technology, Cengage Learning, 2006.
- P. K.Sinha, P. Sinha, Fundamentals of Computers, BPB Publishers, 2007

Paper Code: DSE02

Paper Title: Introduction to Database System (6 Credits, L+T+P = 4+0+2)

This course introduces the basics of database management systems. The course is designed to elaborate how data stores, manage and access in computer systems. It also provides a detailed learning of how to apply query languages to retrieve data from databases and use those data to make logical decisions for real world problems.

On successful completion of the course, students will be able to:

1. Understand the basics of data, database and its applications.
2. Learn about the basic of database models and architectures
3. Understand and use SQL to query, update, and manage a database.
4. Design database based applications based on their requirement
5. Get an idea of emerging databases and their applications to solve real world problems.

Unit	Content	No. of Lectures
1	Database: Introduction to database, relational data model, DBMS architecture, data independence, DBA, database users, end users, front end tools.	14
2	E-R Modeling: Entity types, entity set, attribute and key, relationships, relation types, E- R diagrams, database design using ER diagrams.	14
3	Relational Data Model: Relational model concepts, relational constraints, primary and foreign key, normalization: 1NF, 2NF, 3NF.	14
4	Structured Query Language: SQL queries, create a database table, create relationships between database tables, modify and manage tables, queries, forms, reports, modify, filter and view data.	18
5	Practical 1. Create a database having two tables with the specified fields, to computerize a library system of a Delhi University College. LibraryBooks (Accession number, Title, Author, Department, PurchaseDate, Price) IssuedBooks (Accession number, Borrower)	2 Credits

	<p>a. Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.</p> <p>b. Delete the record of a book titled —Database System Concepts.</p> <p>c. Change the Department of the book titled —Discrete Maths to —CS.</p> <p>d. List all books that belong to —CS department.</p> <p>e. List all books that belong to —CS department and are written by the author —Navathe.</p> <p>f. List all computer books (Department=CS) that have been issued.</p> <p>g. List all books which have a price less than 500 or purchased between —01/01/1999 and —01/01/2004.</p> <p>2. Create Database having tables to store the details of students of Computer Department in college</p> <p>Personal information about Student (College roll number, Name of student, Date of birth, Address, Marks(rounded off to whole number) in percentage at 10 + 2, Phone number) Paper Details (Paper code, Name of the Paper) Student Attendance, Marks in Attendance details)(College roll number, Paper Code, Attendance, Marks in Attendance details)</p> <p>a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.</p> <p>b) Design a query that will return the records (from the second table) along with the name of student from the first table, related to students who have more than 75% attendance and more than 60% marks in paper 2.</p> <p>c) List all students who live in —Delhi and have marks greater than 60 in paper 1.</p> <p>d) Find the total attendance and total marks obtained by each student.</p> <p>e) List the name of the student who has got the highest marks in paper 2.</p> <p>3. Create the following tables and answer the queries given below:</p> <p>Customer (CustID, email, Name, Phone, ReferrerID) Bicycle (BicycleID, DatePurchased, Color, CustID, ModelNo) BicycleModel (ModelNo, Manufacturer, Style) Service (StartDate, BicycleID, EndDate)</p> <p>a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.</p>	
--	--	--

- b) List all the customers who have bicycles manufactured by manufacturer —Honda.
- c) List the bicycles purchased by the customers who have been referred by the customer —C1.
- d) List the manufacturer of red colored bicycles.
- e) List the models of the bicycles given for service.

4. Create the following tables, enter at least 5 records in each table and answer the queries given below.

EMPLOYEE (Person_Name, Street, City) WORKS (Person_Name, Company_Name, Salary) COMPANY (Company_Name, City)

MANAGES (Person_Name, Manager_Name)

- a) Identify primary and foreign keys.
- b) Alter table employee, add a column —email of type varchar(20).
- c) Find the name of all managers who work for both Samba Bank and NCB Bank.
- d) Find the names, street addresses and cities of residence and salary of all employees who work for Samba Bank and earn more than \$10,000.
- e) Find the names of all employees who live in the same city as the company for which they work.
- f) Find the highest salary, lowest salary and average salary paid by each company.
- g) Find the sum of salary and number of employees in each company.
- h) Find the name of the company that pays the highest salary.

5. Create the following tables, enter at least 5 records in each table and answer the queries given below.

Suppliers (SNo, Sname, Status, SCity)

	<p>Parts (PNo, Pname, Colour, Weight, City)</p> <p>Project (JNo, Jname, Jcity)</p> <p>Shipment (Sno, Pno, Jno, Quantity)</p> <p>a) Identify primary and foreign keys.</p> <p>b) Get supplier numbers for suppliers in Paris with status>20.</p> <p>c) Get suppliers details for suppliers who supply part P2. Display the supplier list in increasing order of supplier numbers.</p> <p>d) Get suppliers names for suppliers who do not supply part P2.</p> <p>e) For each shipment get full shipment details, including total shipment weights.</p> <p>f) Get all the shipments where the quantity is in the range 300 to 750 inclusive.</p> <p>g) Get part nos. for parts that either weigh more than 16 pounds or are supplied by suppliers S2, or both.</p> <p>h) Get the names of cities that store more than five red parts.</p> <p>i) Get full details of parts supplied by a supplier in London.</p> <p>j) Get part numbers for part supplied by a supplier in London to a project in London.</p> <p>k) Get the total number of projects supplied by a supplier (say, S1).</p> <p>l) Get the total quantity of a part (say, P1) supplied by a supplier (say, S1).</p>	
--	--	--

Reading list:

- P. Rob, C. Coronel, Database System Concepts by, Cengage Learning India, 2008
- R. Elmsasri, S. Navathe Fundamentals of Database Systems, Pearson Education, Fifth Edition, 2007
- MySQL : Reference Manual

Paper Code: DSE03

Paper Title: Introduction to Programming (6 Credits, L+T+P = 4+0+2)

This course is aimed to provide a basic know-how of programming using C and C++. The course discusses the different fundamental concepts of procedural programming as well as the objective oriented programming paradigm using programming examples.

On successful completion of the course, students will be able to:

1. Understand what is meant by programming and how it is useful while solving problems.
2. Learn about the basic of C and C++
3. Write code, compile and execute programs.
4. Understand the concepts of OOPs and able to model/solve problems using OOPs in different application domain

Unit	Content	No. of Lectures
1	Introduction to C and C++: History of C and C++, Overview of Procedural Programming and Object-Orientation Programming, Using main() function, Compiling and Executing Simple Programs in C++.	5
2	Data Types, Variables, Constants, Operators and Basic I/O: 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, putchar etc), Formatted and Console I/O (printf(), scanf(), cin, cout), Using Basic Header Files (stdio.h, iostream.h, conio.h etc)	10
3	Expressions, Conditional Statements and Iterative Statements: 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)	10
4	Functions and Arrays Utility of functions, Call by Value, Call by Reference, Functions	10

	<p>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.</p> <p>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</p>	
5	<p>Derived Data Types (Structures and Unions): 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.</p>	5
6	<p>File I/O, Preprocessor Directives: 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</p>	8
7	<p>Using Classes in C++ 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.</p>	8
8	<p>Inheritance and Polymorphism Introduction to Inheritance and Polymorphism</p>	4
	<p>Practical</p> <ol style="list-style-type: none"> 1. Write a program to find the greatest of three numbers. 2. Write a program to find gross salary of a person 	2 Credits

	<p>3. Write a program to find the grade of a student given his marks.</p> <p>4. Write a program to find the divisor or factorial of a given number.</p> <p>5. Write a program to print the first ten natural numbers.</p> <p>6. Write a program to print the first ten even and odd numbers.</p> <p>7. Write a program to find the grade of a list of students given their marks.</p> <p>8. Create Matrix class. Write a menu-driven program to perform following Matrix operations (2-D array implementation): a) Sum b) Difference c) Product d) Transpose.</p>	
--	---	--

Reading List:

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

Paper Code: DSE04

Paper Title: Computer Networks and Internet Technologies (6 Credits, L+T+P = 4+0+2)

This course is aimed to provide an overview of basic networking technologies and their usage in various application domains. The course introduces the notion of computer networks, models, and

different technologies used to make networking viable. The course also includes some basics of web-design technology and framework uses in web pages.

On successful completion of the course, students will be able to:

1. Understand what is meant by a computer network and its application in various domains.
2. Learn about different components and technologies uses for designing of a computer network
3. Able to apply knowledge of layered architecture in building LAN.
4. Get idea about various technology and frameworks for web-design
5. Apply the theoretical knowledge to design web-pages which can run over network for a given assignment

Unit	Content	No. of Lectures
1	Computer Networks Introduction to computer network, data communication, components of data communication, data transmission mode, data communication measurement, LAN, MAN, WAN, wireless LAN, internet, intranet, extranet.	6
2	Network Models Client/ server network and Peer-to-peer network, OSI, TCP/IP, 8L layers and functionalities.	8
3	Transmission Media Introduction, Guided Media: Twisted pair, Coaxial cable, Optical fiber. Unguided media: Microwave, Radio frequency propagation, Satellite.	4
4	LAN Topologies Ring, bus, star, mesh and tree topologies	2
5	Network Devices NIC, repeaters, hub, bridge, switch, gateway and router.	2
6	Internet Terms Web page, Home page, website, internet browsers, URL, Hypertext, ISP, Web server, download and upload, online and offline.	2
7	Internet Applications www, telnet, ftp, e-mail, social networks, search engines, Video	6

	Conferencing, e-Commerce, m-Commerce, VOIP, blogs.														
8	<p>Introduction to Web Design</p> <p>Introduction to hypertext markup language (html) Document type definition, creating web pages, lists, hyperlinks, tables, web forms, inserting images, frames, hosting options and domain name registration. Customized Features: Cascading style sheet (css) for text formatting and other manipulations.</p>	16													
9	<p>JavaScript Fundamentals</p> <p>Data types and variables, functions, methods and events, controlling program flow, JavaScript object model, built-in objects and operators.</p>	14													
	<p>Practical</p> <p>Practical exercises based on concepts listed in theory using HTML.</p> <ol style="list-style-type: none"> 1. Create HTML document with following formatting – Bold, Italics, Underline, Colors, Headings, Title, Font and Font Width, Background, Paragraph, Line Brakes, Horizontal Line, Blinking text as well as marquee text. 2. Create HTML document with Ordered and Unordered lists, Inserting Images, Internal and External linking. 3. Create HTML document with Table: <table border="1" data-bbox="321 1360 1276 1602"> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td colspan="3"></td> <td colspan="2" rowspan="2">Some Image Here</td> </tr> <tr> <td colspan="3"></td> </tr> </table> 4. Create Form with Input Type, Select and Text Area in HTML. 5. Create an HTML containing Roll No., student's name and Grades in a tabular form. 6. Create an HTML document (having two frames) which will appear as 									Some Image Here					2 Credits
			Some Image Here												

follows:

About	This frame would show the contents according to the link clicked by the user on the left frame.
Department 1	
Department 2	
Department 3	

7. Create an HTML document containing horizontal frames as follows:

Department Names (could be along with Logos)
Contents according to the Link clicked

8. Create a website of 6 – 7 pages with different effects as mentioned in above problems.

9. Create HTML documents (having multiple frames) in the following three formats:

Frame 1 Frame 2

Frame 1	
Frame 2	Frame 3

10. Create a form using HTML which has the following types of controls:

- I. Text Box
- II. Option/radio buttons
- III. Check boxes

IV. Reset and Submit buttons

Subscribe to XYZ News Magazine and Emails

Interested in receiving daily small updates of all latest News? Well, now you can. And best of all, it is free! Just fill out this form and submit it by clicking the "send it In" button. We will put you on our mailing list and you will receive your first email in 3-5 days.

Please fill the following boxes to help us send the emails and our news letter:

First Name:

Last Name:

Business:

We must have a correct e-mail address to send you the news letter.

Email:

How did you hear about XYZ News Magazine and Emails?

Here on the Web In a magazine Television Other

Would you like to be on our regular mailing list?

Yes, we love junk emails

List of Practicals using Javascript :

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

Reading List

- Andrew S. Tanenbaum, David J. Wetherall Computer Networks (5th Edition),PHI, 2010
- B. A. Forouzan, Data Communication and Networking , TMH,2003.
- D.R. Brooks, An Introduction to HTML and Javascript for Scientists and Engineers, Springer W. Willard,2009
- HTML A Beginner's Guide, Tata McGraw-Hill Education, 2009
- J. A. Ramalho, Learn Advanced HTML 4.0 with DHTML, BPB Publications, 2007

Paper Code: DSE05

Paper Title: Multimedia and Applications (6 Credits, L+T+P = 4+0+2)

This course is aimed to provide an overview Multimedia Technologies and its applications in various domain. As multi-media become intrinsic part of today's world, this course includes the different multimedia. The course also includes some basics of web-design technology and framework uses in web pages.

On successful completion of the course, students will be able to:

1. Understand what is meant by a computer network and its application in various domains.
2. Learn about different components and technologies uses for designing of a computer network
3. Get idea about various technology and frameworks for web-design
4. Apply the theoretical knowledge to design web-pages which can run over network for a given assignment

Unit	Content	No. of Lectures
1	Multimedia Introduction to multimedia, components, uses of multimedia, multimedia applications, virtual reality.	6
2	Text Fonts & Faces, Using Text in Multimedia, Font Editing & Design Tools, Hypermedia & Hypertext	4
3	Images Still Images – bitmaps, vector drawing, 3D drawing & rendering, natural light & colors, computerized colors, color palettes, image file formats.	6
4	Sound Digital Audio, MIDI Audio, MIDI vs Digital Audio, Audio File Formats.	6
5	Video How video works, analog video, digital video, video file formats, video shooting and editing.	8
6	Animation Principle of animations, animation techniques, animation file formats.	10

7	<p>Internet and Multimedia www and HTML, multimedia on the web – web servers, web browsers, web page makers and site builders.</p>	6
8	<p>Making Multimedia Stages of a multimedia project, Requirements to make good multimedia, Multimedia Hardware - Macintosh and Windows production Platforms, Hardware peripherals - Connections, Memory and storage devices, Multimedia software and Authoring tools.</p>	14
	<p>Practical</p> <p>Practical exercises based on concepts listed in theory using Flash/ GIMP/ PhotoShop/ Animation</p> <p>Tools/ Image Editors/ Video Editors.</p> <p>Optional</p> <p>Implement the followings using Flash-</p> <ol style="list-style-type: none"> 1. Create an animation using the tools panel and the properties panel to draw the following – Line, pe , oval, circle, rectangle , square, pencil , brush , lasso tool 2. Create an animation using a text tool to set the font, size , color etc. 3. Create an animation using Free transform tool that should use followings-Move Objects Skew Objects Stretch Objects Rotate Objects Stretch Objects while maintaining proportion Rotate Objects after relocating the center dot 4. Create an animation using layers having following features-Insert layer, Delete layer, guide layer, Mask layer. 5. Modify the document (changing background color etc.)using the following tools Eraser tool Hand tool Ink bottle tool Zoom tool Paint Bucket tool Eyedropper tool 	2 Credits

	<p>6. Create an animation for a bus car race in which both start from the same point and the car wins the race.</p> <p>7. Create an animation in which text Hello gets converted into GoodBye (using motion/shape tweening).</p> <p>8. Create an animation having five images having fade-in fade-out effect.</p> <p>9. Create an scene to show the sunrise (using multiple layers and motion tweening)</p> <p>10. Create an animation to show the ripple effect.</p> <p>11. Create an animation (using Shape tweening and shape hints) for transforming one shape into another.</p> <p>12. Create an animation for bouncing balls (you may use a motion guide layer).</p>	
--	--	--

Reading List:

- Tay Vaughan, —Multimedia: Making it work, TMH, Eighth edition.2011
- Ralf Steinmetz and Klara Naharstedt, —Multimedia: Computing, Communications Applications, Pearson.2012
- Keyes, ``Multimedia Handbook'', TMH,2000.
- K. Andleigh and K. Thakkar, —Multimedia System Designl, PHI.2013

Paper Code: DSE06

Paper Title: Programming in Python (6 Credits, L+T+P = 4+0+2)

This course is aimed to provide an overview on programming basics and paradigm. The course helps to develop logic for problem solving and implement solutions using Python programming language. It discusses the basics of python looping, control statements and string manipulations along with the basics of OOPs and data structures.

On successful completion of the course, students will be able to:

1. Understand the basics of problem solving and about programming paradigms
2. Define and demonstrate the use of built-in data structures
3. Application of OOPs concepts to model real word problems
4. Design and implement a program to solve a real world problem

Unit	Content	No. of Lectures
1	Planning the Computer Program Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation.	4
2	Techniques of Problem Solving Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming.	6
3	Overview of Programming Structure of a Python Program, Elements of Python	4
4	Introduction to Python Python Interpreter, Using Python as calculator, Python shell, Indentation. Atoms, Identifiers and keywords, Literals, Strings, Operators (Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bit wise operator, Increment or Decrement operator)	6
5	Creating Python Programs Input and Output Statements, Control statements (Branching, Looping, Conditional Statement, Exit function, Difference between break, continue and pass.), Defining Functions, default arguments, Errors and Exceptions.	8
6	Iteration and Recursion Conditional execution, Alternative execution, Nested conditionals, The return statement, Recursion, Stack diagrams for recursive functions, Multiple assignment, The while statement, Tables, Two-dimensional tables.	8
7	Iteration and Recursion Conditional execution, Alternative execution, Nested conditionals, The return statement, Recursion, Stack diagrams for recursive functions, Multiple assignment, The while statement, Tables, Two-dimensional tables.	8
8	Object Oriented Programming Introduction to Classes, Objects and Methods, Standard Libraries.	4
9	Data Structures Arrays, list, set, stacks and queues.	6

10	Searching and Sorting Linear and Binary Search, Bubble, Selection and Insertion sorting.	6
	<p>Practical</p> <ol style="list-style-type: none"> 1. Using for loop, print a table of Celsius/Fahrenheit equivalences. Let c be the Celsius temperatures ranging from 0 to 100, for each value of c, print the corresponding Fahrenheit temperature. 2. Using a while loop, produce a table of sines, cosines and tangents. Make a variable x in range from 0 to 10 in steps of 0.2. For each value of x, print the value of sin(x), cos(x) and tan(x). 3. Write a program that reads an integer value and prints —leap year or —not a leap year. 4. Write a program that takes a positive integer n and then produces n lines of output shown as follows. For example enter a size: 5 <pre style="margin-left: 20px;">* ** *** **** *****</pre> 5. Write a function that takes an integer <code>_n</code> as input and calculates the value of $1 + 1/1! + 1/2! + 1/3! + \dots + 1/n$ 6. Write a function that takes an integer input and calculates the factorial of that number. 7. Write a function that takes a string input and checks if it's a palindrome or not. 8. Write a list function to convert a string into a list, as in list (abc) gives [a, b, c]. 9. Write a program to generate Fibonacci series. 	2 Credits

	<p>10. Write a program to check whether the input number is even or odd.</p> <p>11. Write a program to compare three numbers and print the largest one.</p> <p>12. Write a program to print factors of a given number.</p> <p>13. Write a method to calculate GCD of two numbers.</p> <p>14. Write a program to create Stack Class and implement all its methods. (Use Lists).</p> <p>15. Write a program to create Queue Class and implement all its methods. (Use Lists)</p> <p>16. Write a program to implement linear and binary search on lists.</p> <p>17. Write a program to sort a list using insertion sort and bubble sort and selection sort.</p>	
--	--	--

Reading List:

- T. Budd, Exploring Python, TMH, 1st Ed, 2011
- How to think like a computer scientist : learning with Python / Allen Downey, Jeffrey Elkner, Chris Meyers. 1st Edition – Freely available online.2012
- <http://docs.python.org/3/tutorial/index.html>
- <http://interactivepython.org/courselib/static/pythonds>

Paper Code: DSE07

Paper Title: Programming in Visual Basic / Gambas (6 Credits, L+T+P = 4+0+2)

This course is aimed to cover visual basic programming skills required for modern software development. It discusses the various advantages of visual basic, basic understanding of database access and management using data controls.

On successful completion of the course, students will be able to:

1. Understand the basics of various tools of a visual environment such as command, menus and toolbars and their utility.
2. Learn to design GUI using forms, dialogs, and controls.
3. Understand the connectivity between VB and other popular database management systems such as MS-ACCESS and MySQL.
4. Design applications to solve a real world problem

Unit	Content	No. of Lectures
1	GUI Environment Introduction to graphical user interface (GUI), programming language (procedural, object oriented, event driven), the GUI environment, compiling, debugging, and running the programs.	4
2	Controls Introduction to controls textboxes, frames, check boxes, option buttons, images, setting borders and styles, the shape control, the line control, working with multiple controls and their properties, designing the user interface, keyboard access, tab controls, default & cancel property, coding for controls.	8
3	Operations Data types, constants, named & intrinsic, declaring variables, scope of variables, val function, arithmetic operations, formatting data.	4
4	Decision Making If statement, comparing strings, compound conditions (and, or, not), nested if statements, case structure, using if statements with option buttons & check boxes, displaying message in message box, testing whether input is valid or not.	8
5	Modular Programming Menus, sub-procedures and sub-functions defining / creating and modifying a menu, using common dialog box, creating a new subprocedure, passing variables to procedures, passing argument by value or by reference, writing a function/ procedure.	6
6	Forms Handling Multiple forms creating, adding, removing forms in project, hide, show method, load, unload statement, me keyword, referring to objects on a different forms	6
7	Iteration Handling Do/loops, for/next loops, using msgbox function, using string function	4
8	Arrays and Grouped Data Control Arrays - 1-dimension arrays, initializing an array using for each, user-defined data types, accessing information with user-defined data types, using list boxes with array, two dimensional arrays. lists, loops and printing list boxes & combo boxes, filling the list using property window / add item method, clear method, list box properties, removing an item from a list, list box/ combo box operations.	10

9	<p>Database Connectivity Database connectivity of forms with back end tools like mysql, populating the data in text boxes, list boxes etc. searching for data in database using forms. Updating/ editing of data based on a criterion.</p>	10
	<p>Practical</p> <ol style="list-style-type: none"> 1. Print a table of numbers from 5 to 15 and their squares and Cubes. 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. Complete the amount in the account at the end of each year for n years. 6. Read n numbers. Count the number of negative numbers, positive numbers and zeros in the list. 7. Read n numbers. Count the number of negative numbers, positive numbers and zeroes in the list.use arrays. 8. Read a single dimension array. Find the sum and average of these numbers. 9. Read a two dimensional array. Find the sum of two 2D Array. 10. Create a database Employee and Make a form in VB 6.0 to allow data entry to Employee Form with the following command buttons: <p>Employee Form</p> <p>Employee Name: <input type="text" value="Employee Name"/> <input type="button" value="NEXT"/></p> <p>Employee Id: <input type="text"/></p>	2 Credits

Reading List:

- Reference: Programming in Visual Basic 6.0 by Julia Case Bradley, Anita C. Millispangh (Tata Mcgraw Hill Edition 2000 (Fourteenth Reprint 2004))

Paper Code: DSE08

Paper Title: Information Security and Cyber Laws (6 Credits, L+T+P = 4+0+2)

This course is aimed to provide a basic overview of Information Security and its anatomy, and basics of Cyber Law as per IT act.

On successful completion of the course, students will be able to:

1. Understand the various threats and vulnerabilities of computer systems including software and hardware components.

2. Understand different types attack and their consequences
3. Study various existing safety tools and analyse their issues
4. Design countermeasures/solutions against common Information Security Attack

Unit	Content	No. of Lectures
1	Course Introduction Computer network as a threat, hardware vulnerability, software vulnerability, importance of data security	8
2	Digital Crime Overview of digital crime, criminology of computer crime.	4
3	Information Gathering Techniques Tools of the attacker, information and cyber warfare, scanning and spoofing, password cracking, malicious software, session hijacking	8
4	Risk Analysis and Threat Risk analysis, process, key principles of conventional computer security, security policies, authentication, data protection, access control, internal vs external threat, security assurance, passwords, authentication, and access control, computer forensics and incident response	10
5	Introduction to Cryptography and Applications Important terms, Threat, Flaw, Vulnerability, Exploit, Attack, Ciphers, Codes, Caesar Cipher, Rail-Fence Cipher, Public key cryptography (Definitions only), Private key cryptography (Definition and Example)	10
6	Safety Tools and Issues Firewalls, logging and intrusion detection systems, Windows and windows XP / NT security, Unix/Linux security, ethics of hacking and cracking	10
7	Cyber laws to be covered as per IT 2008 <ul style="list-style-type: none"> ● Chapter 1: Definitions ● Chapter 2: Digital Signature And Electronic Signature ● [Section 43] Penalty and Compensation for damage to computer, computer system, etc. ● [Section 65] Tampering with Computer Source Documents ● [Section 66 A] Punishment for sending offensive messages through communication service, etc. ● [Section 66 B] Punishments for dishonestly receiving stolen computer resource or communication device ● [Section 66C] Punishment for identity theft 	10

	<ul style="list-style-type: none"> ● [Section 66D] Punishment for cheating by personation by using computer resource ● [Section 66E] Punishment for violation of privacy ● [Section 66F] Punishment for cyber terrorism ● [Section 67] Punishment for publishing or transmitting obscene material in electronic form ● [Section 67A] Punishment for publishing or transmitting of material containing sexually explicit act, etc. in electronic form ● [Section 67B] Punishment for publishing or transmitting of material depicting children in sexually explicit act, etc. in electronic form ● [Section 72] Breach of confidentiality and privacy 	
	<p>Practical</p> <ol style="list-style-type: none"> 1. Demonstrate the use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois 2. Use of Password cracking tools : John the Ripper, Ophcrack. Verify the strength of passwords using these tools. 3. Perform encryption and decryption of Caesar cipher. Write a script for performing these operations. 4. Perform encryption and decryption of a Rail fence cipher. Write a script for performing these operations. 5. Use nmap/zenmap to analyse a remote machine. 6. Use Burp proxy to capture and modify the message. 7. Demonstrate sending of a protected word document. 8. Demonstrate sending of a digitally signed document. 9. Demonstrate sending of a protected worksheet. 10. Demonstrate use of steganography tools. 11. Demonstrate use of gpg utility for signing and encrypting purposes. 	2 Credits

Reading List:

- M. Merkow, J. Breithaupt, Information Security Principles and Practices, Pearson Education.2005
- G.R.F. Snyder, T. Pardoe, Network Security, Cengage Learning, 2010
- A. Basta, W.Halton, Computer Security: Concepts, Issues and Implementation, Cengage Learning India, 2008

Paper Code: DSE09

Paper Title: Web and E-Commerce Technologies (6 Credits, L+T+P = 4+0+2)

This course is aimed to provide a basic overview of Information Security and its anatomy, and basics of Cyber Law as per IT act.

On successful completion of the course, students will be able to:

1. Understand the basics of E-commerce and its impact on business growth.
2. Understand the basic concepts and technologies used in the field of E-commerce and its management.
3. Study various security threats and analyse their impacts on information exchange in e-commerce environments
4. Design models for building an efficient E-commerce platform

Unit	Content	No. of Lectures
1	An introduction to Electronic commerce: What is E-Commerce (Introduction And Definition), Main activities E-Commerce, Goals of E-Commerce, Technical Components of ECommerce, Functions of E-Commerce, Advantages and disadvantages of E-Commerce, Scope of E-Commerce, Electronic Commerce Applications, 9 Electronic Commerce and Electronic Business(C2C)(C2G,G2G, B2G, B2P, B2A, P2P, B2A, C2A, B2B, B2C)	10
2	The Internet and WWW: Evolution of Internet, Domain Names and Internet Organization (.edu, .com, .mil, .gov, .net etc.) , Types of Network, Internet Service Provider, World Wide Web, Internet & Extranet, Role of Internet in B2B Application, building own website, Cost, Time, Reach, Registering a Domain Name, Web promotion, Target email, Baner, Exchange, Shopping Bots.	10
3	Internet Security Secure Transaction, Computer Monitoring, Privacy on Internet, Corporate Email privacy, Computer Crime(Laws , Types of Crimes), Threats, Attack on Computer System, Software Packages for privacy, Hacking, Computer Virus(How it spreads, Virus problem, virus protection, Encryption and Decryption, Secret key Cryptography, DES, Public Key Encryption, RSA, Authorisation and Authentication, Firewall, Digital Signature(How it Works)	10
4	Electronic Data Exchange [10 Lectures] Introduction, Concepts of EDI and Limitation, Applications of EDI, Disadvantages of EDI, EDI model, Electronic Payment System: Introduction, Types of Electronic Payment System, Payment Types, Value Exchange System, Credit Card System, Electronic Fund Transfer, Paperless bill, Modern Payment Cash, Electronic Cash	10
5	Planning for Electronic Commerce Planning Electronic Commerce initiates, Linking objectives to business strategies, Measuring cost objectives, Comparing benefits to Costs, Strategies for developing	10

	electronic commerce web sites	
6	<p>Internet Marketing The PROS and CONS of online shopping, The cons of online shopping, Justify an Internet business, Internet marketing techniques, The E-cycle of Internet marketing, Personalization e-commerce.</p>	10
	<p>Practical Web and E- Commerce Technologies LAB (based on the following topics): HyperText Markup Language (HTML): structural setup; page layout; text manipulation; special characters; images; links. Intermediate: image maps; tables; frames, forms; meta tags; web forms. Cascading Style Sheets (CSS): embedding/linking; HTML element selectors; classes; ID selectors, text manipulation; background; borders and spacing; layout; context selectors and grouping, pseudo classes; pseudo-elements. JavaScript: writing your first script; creating HTML tags; user input and output; loops and tables; payroll calculator, forms and text fields; validating an email address; radio buttons; check boxes; self-grading tests, image rollovers; slide shows; real-time clock; controllable clock; working with cookies. Perl/CGI 10: sample Perl operations; random numbers; lists; dealing four poker hands; time manipulation; subroutines, hash tables; files; string matching, CGI; registration lists; surveys. SQL and regular expressions: Regular expressions: basics; repeating; positioning. Beginner: select; where; order by; insert; update; delete, like; between; in; distinct; group by; aliases; aggregate functions; create table; alter table; drop table, nested selects; SoundEx; join; deterministic functions; non-deterministic functions. ASP structural setup: response.write; retrieving from forms; retrieving from querystring; variables; control constructs; subroutines and functions; session state; application variables; server variables; debugging, reading and writing cookies; server-side includes; response object methods; VBScript functions; error handling; debugging, browser details; CDONTS; files; output from a recordset; global.asa; setup instructions for using IIS and ASP. Flash 3 Create Flash movies of moving and interactive objects.</p>	2 Credits

Reading List:

- G.S.V.Murthy, E-Commerce Concepts, Models, Strategies- :- Himalaya Publishing House, 2011.
- Kamlesh K Bajaj and Debjani Nag , E- Commerce , 2005.
- Gray P. Schneider , Electronic commerce, International Student Edition, 2011,
- Henry Chan, Raymond Lee, Tharam Dillon, Elizabeth Chang E-commerce, Fundamentals And Applications, Wiely Student Edition, 2011

Paper Code: DSE10

Paper Title: Information Security (6 Credits, L+T+P = 4+0+2)

The objective of this course is to understand the basics of Cryptography and Network Security. To learn about how to maintain the Confidentiality, Integrity and Availability of data. To understand various protocols for network security to protect against the threats in the networks.

On successful completion of the course, students will be able to:

1. Secure a message over an insecure channel by various means.
2. Perform research in the emerging areas of cryptography and network Security.
3. Implement various networking protocols.
4. Identify, explain, and apply cryptographic techniques like key management, digital signatures, digital certificates, and a Public-Key Infrastructure (PKI) to various disciplines in information science.

Unit	Content	No. of Lectures
1	Introduction: Security, Attacks, Computer Criminals, Security Services, Security Mechanisms.	5
2	Cryptography: Substitution ciphers, Transpositions Cipher, Confusion, Diffusion, Symmetric, Asymmetric Encryption. DES Modes of DES, Uses of Encryption, Hash function, Key exchange, Digital Signatures, Digital Certificates.	10
3	Program Security: Secure programs, Non malicious Program errors, Malicious Code Virus, Trap doors, Salami attacks, Covert channels, Control against program	10
4	Threats: Protection in OS: Memory and Address Protection, Access control, File Protection, User Authentication.	10
5	Database Security: Requirements, Reliability, Integrity, Sensitive data, Inference, Multilevel Security.	5
6	Security in Networks: Threats in Networks, Security Controls, firewalls, Intrusion detection systems, Securee-mails	10
7	Administering Security: Security Planning, Risk Analysis, Organisational Security Policy, Physical Security, Ethical issues in Security: Protecting Programs and data. Information and law.	10
	Practical: <ol style="list-style-type: none"> 1. Demonstrate the use Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois 2. Use of Password cracking tools: Johnthe Ripper, Ophcrack. Verify the strength of passwords using these tools. 3. Perform encryption and decryption of Caesar cipher. Write a script for performing these 	2 Credits

	<p>operations.</p> <ol style="list-style-type: none"> 4. Perform encryption and decryption of a Rail fence cipher. Write a script for performing these Operations. 5. Use nmap/zenmap to analyse a remote machine. 6. Use Burp proxy to capture and modify the message. 7. Demonstrate sending of a protected word document. 8. Demonstrate sending of a digitally signed document. 9. Demonstrate sending of a protected worksheet. 10. Demonstrate use of steganography tools. 11. Demonstrate use of gpg utility for signing and encrypting purposes. 	
--	--	--

Reading list:

- C.P.Pfleeger, S. L.Pfleeger; Security In Computing, Prentice Hall of India, 2006
- W.Stallings; Network Security Essentials: Applications and Standards,4/E,2010

Paper Code: DSE11

Paper Title: Network Programming (6 Credits, L+T+P = 4+0+2)

The objective of this course is to implement computer network protocols using socket programming. The course covers the Internet protocol suite (e.g. IP, TCP, UDP, ICMP, and FTP) and client-server design (e.g. connectionless, connection oriented, multiprotocol).

On successful completion of the course, students will be able to:

1. Demonstrate mastery of main protocols comprising the Internet.
2. Develop skills in network programming techniques.
3. Implement network services that communicate through the Internet.
4. Apply the client-server model in networking applications.
5. Practice networking commands available through the operating systems.

Unit	Content	No. of Lectures
1	Transport Layer Protocols: TCP, UDP, SCTP	10
2	Socket Programming: Socket Introduction, TCPSockets, TCP Client/Server Example	15
3	Signal handling: I/O multiplexing using sockets, Socket Options, UDP Sockets, UDP client/server example;	15

4	Address lookup using sockets. Network Applications: Remote logging, Email, WWW and HTTP.	10
5	LAN administration: Linux and TCP/IP networking, Network Management and Debugging	10
	<p>Practical:</p> <ol style="list-style-type: none"> 1. Write a socket programming to implement a simple client and server that communicate over the network. 2. Use socket programming to implement a simple client and server that communicate over the network and implement a simple Client Server Communication The client opens a session with the server on a predefined port and then supplies a line of Text Messages. The server outputs the Text Message on standard output, and also keeps on running to receive more Text Messages. When the client closes the session by choosing the Exit option, the connection to the server should be gracefully terminated. When the server is terminated by pressing control+C, the server should also gracefully release the open socket. 3. Use of network tools like wireshark, tcpdump to capture network traffic for analysis. 4. Use of network simulators like NS-2, Omnet++, etc. 5. Practical exercises based on concepts listed in theory. 	2 Credits

Reading list:

- W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, Unix Network Programming, The sockets Networking API, Vol. 1, 3rd Edition, PHI, 2003
- B.A. Forouzan: Data Communications and Networking, 4th edition, THM Publishing Company Ltd., 2003
- Nemeth Synder & Hein, Linux Administration Handbook, Pearson Education, 2nd Edition, 2010
- R. Stevens, Unix Network Programming, PHI 2nd Edition, 1990

Paper Code: DSE12

Paper Title: Microprocessor (6 Credits, L+T+P = 4+0+2)

This subject deals with the basics of microprocessors, their architecture, internal organization and their functions, interfacing an external device with the processors/controllers.

On successful completion of the course, students will be able to:

1. Recall and apply a basic concept of digital fundamentals to Microprocessor based personal computer systems.
2. Identify a detailed structure of the Microprocessor.
3. Illustrate how the different peripherals are interfaced with microprocessors.
4. Train their practical knowledge through laboratory experiments.

Unit	Content	No. of Lectures
1	Microprocessor architecture: Internal architecture, system bus architecture, memory and I/O interfaces.	15
2	Microprocessor programming: Register Organization, instruction formats, assembly language programming.	20
3	Interfacing: Memory address decoding, cache memory and cache controllers, I/O interface, keyboard, display, timer, interrupt controller, DMA controller, video controllers, communication interfaces.	25
	Practical: <ol style="list-style-type: none"> 1. Write a program for 32-bit binary division and multiplication. 2. Write a program for 32-bit BCD addition and subtraction. 3. Write a program for Linear search and binary search. 4. Write a program to add and subtract two arrays. 5. Write a program for binary to ASCII conversion. 6. Write a program for ascii to binary conversion. 	2 Credits

Reading list:

- Barry B.Brey: The Intel Microprocessors: Architecture, Programming and Interfacing. Pearson Education, Sixth Edition, 2009.
- Walter A Triebel, Avtar Singh: The 8088 and 8086 Microprocessors Programming, Interfacing, Software, Hardware, and Applications. PHI, Fourth Edition 2005.

Paper Code: DSE13

Paper Title: Computational Linguistics (6 Credits, L+T+P = 4+0+2)

This course will help the students to learn Natural Language Processing - problems and perspectives, basic corpus linguistics. They will be able to map the appropriate processing technique to a problem and implement the technique. They will also be able to demonstrate required design skills for large collection sets.

On successful completion of the course, students will be able to:

1. Manage some of the main topics in Natural Language Processing placing particular emphasis on advanced empirical methods used for linguistic analysis.
2. Analyze a problem, and identify and define the computing as well as the linguistics requirements appropriate to its solution.
3. Design, implement, and evaluate a computational linguistics-based system, process, component, or program to meet desired text processing needs.

Unit	Content	No. of Lectures
1	Introduction to Computers in linguistics and Natural Language Processing The nature and use of text corpora	4
2	Introduction to python programming and NLTK, Regular expressions Pattern matching Corpus search and counting	8
3	Regular languages Finite-state automata Operations and closure properties Pumping Lemma, Finite-state linguistics Transducers Morphological analysis	8
4	N-grams Language modeling Smoothing Evaluation	8
5	Part-of-Speech Tagging, Word classes and tagsets Rule-based and stochastic POS tagging Hidden Markov Models Evaluation	8
6	Word meaning Semantic ambiguity Semantic relations Semantic roles	8
7	Computational lexical semantics, (Un)supervised word sense disambiguation Classifiers Vector-space semantics	8
	Practical: 1. Practical exercises based on concepts listed in theory.	2 Credits

Reading list:

- Jurafsky, D. and J. H. Martin, Speech and language processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition(2ndEdition). Prentice-Hall, 2008.
- Charniak,E.: Statistical Language Learning. The MIT Press. 1996.
- J.Allen: Natural Language Understanding. Benjamin/Cummins. 1995.

Paper Code: DSE14

Paper Title: Digital Image Processing (6 Credits L+T+P = 4+0+2)

The objective of this course is to introduce the concepts of image processing and basic analytical methods to be used in image processing. To familiarize students with image enhancement and restoration techniques, to explain different image compression techniques. To introduce segmentation and morphological processing techniques.

On successful completion of the course, students will be able to:

1. Explain the fundamentals of digital image and its processing.
2. Perform image enhancement techniques in spatial and frequency domain.
3. Elucidate the mathematical modeling of image restoration and compression.
4. Apply the concept of image segmentation.
5. Describe object detection and recognition techniques.

Unit	Content	No. of Lectures
1	Introduction: Light, Brightness adaption and discrimination, Pixels, coordinate conventions, Imaging Geometry, Perspective Projection, Spatial Domain Filtering, sampling and quantization.	6
2	Spatial Domain Filtering: Intensity transformations, contrast stretching, histogram equalization, Correlation and convolution, Smoothing filters, sharpening filters, gradient and Laplacian.	7
3	Filtering in the Frequency domain: Hotelling Transform, Fourier Transforms and properties, FFT(Decimation in Frequency and Decimation in Time Techniques), Convolution, Correlation, 2-D sampling, Discrete Cosine Transform, Frequency domain filtering.	8
4	Image Restoration: Basic Framework, Interactive Restoration, Image deformation and geometric transformations, image morphing, Restoration techniques, Noise characterization, Noise restoration filters, Adaptive filters, Linear, Position invariant degradations, Estimation of Degradation functions, Restoration from projections.	8
5	Image Compression: Encoder-Decoder model, Types of redundancies, Lossy and Lossless compression, Entropy of an information source, Shannon's 1st Theorem, Huffman Coding, Arithmetic Coding, Golomb Coding, LZW coding, Transform Coding, Sub-image size selection, blocking artifacts, DCT implementation using FFT, Run length coding, FAX compression (CCITT Group-3 and Group-4), Symbol-based coding, JBIG-2, Bit-plane encoding, Bit-allocation, Zonal Coding, Threshold Coding, JPEG, Lossless predictive coding, Lossy predictive coding, Motion Compensation	10
6	Wavelet based Image Compression: Expansion of functions, Multi-resolution analysis, Scaling functions, MRA refinement equation, Wavelet series expansion, Discrete Wavelet Transform(DWT), Continuous Wavelet Transform, Fast Wavelet Transform, 2-D wavelet Transform, JPEG-2000 encoding, Digital Image Watermarking.	5
7	Morphological Image Processing: Basics, SE, Erosion, Dilation, Opening, Closing, Hit-or-Miss Transform, Boundary Detection, Holefilling, Connected components, convex hull, thinning, thickening, skeletons, pruning, Geodesic Dilation, Erosion, Reconstruction by dilation and erosion.	7
8	Image Segmentation: Boundary detection based techniques, Point, line detection, Edge detection, Edge linking, local processing, regional processing, Hough transform, Thresholding, Iterative thresholding, Otsu's Method, Moving averages, Multivariable thresholding, Region-based segmentation, Watershed Algorithm, Use of motion in segmentation	9

	<p>Practical:</p> <ol style="list-style-type: none"> 1. Write program to read and display digital image using MATLAB or SCILAB <ol style="list-style-type: none"> a. Become familiar with SCILAB/MATLAB Basic commands. b. Read and display image in SCILAB/MATLAB. c. Resize given image. d. Convert given color image into gray-scale image. e. Convert given color/gray-scale image into black & white image. f. Draw image profile. g. Separate color image in three RG&B planes. h. Create a color image using R, G and B three separate planes. i. Flow control and LOOP in SCILAB. j. Write given 2-D data in an image file. 2. To write and execute image processing programs using point processing method <ol style="list-style-type: none"> a. Obtain a Negative image. b. Obtain Flip image . c. Thresholding. d. Contrast stretching. 3. To write and execute programs for image arithmetic operations <ol style="list-style-type: none"> a. Addition of two images. b. Subtract one image from other image. c. Calculate mean value of image. d. Different Brightness by changing mean value. 4. To write and execute programs for image logical operations <ol style="list-style-type: none"> a. AND operation between two images. b. OR operation between two images. c. Calculate intersection of two images. d. Watermarking using EX-OR operation. e. NOT operation(Negative image). 5. To write a program for histogram calculation and equalization using <ol style="list-style-type: none"> a. Standard MATLAB function b. Program without using standard MATLAB functions c. C Program 6. To write and execute program for geometric transformation of image <ol style="list-style-type: none"> a. Translation b. Scaling c. Rotation d. Shrinking e. Zooming 7. To understand various image noise models and to write programs for <ol style="list-style-type: none"> a. image restoration b. Remove Salt And Pepper Noise c. Minimize Gaussian noise d. Median filter and Weiner filter 8. Write and execute programs to remove noise using spatial filters <ol style="list-style-type: none"> a. Understand 1-D and 2-D convolution process b. Use 3x3 Mask for low pass filter and high pass filter 9. Write and execute programs for image frequency domain filtering <ol style="list-style-type: none"> a. Apply FFT on given image b. Perform low pass and high pass filtering in frequency domain c. Apply IFFT to reconstruct image 	<p>2 Credits</p>
--	--	------------------

	<p>10. Write a program in C and MATLAB/SCILAB for edge detection using different edge detection mask</p> <p>11. Write and execute programs for image morphological operations erosion and dilation.</p> <p>12. To write and execute a program for wavelet transform on a given image and perform inverse wavelet transform to reconstruct image.</p>	
--	--	--

Reading list:

- RC Gonzalez, RE Woods, Digital Image Processing, 3rd Edition, Pearson Education. 2008.
- AK Jain, Fundamentals of Digital image Processing, Prentice Hall of India. 1989.
- KR Castleman, Digital Image Processing, Pearson Education. 1996
- Schalkoff, Digital Image Processing and Computer Vision, John Wiley and Sons. 1989.
- Rafael C. Gonzalez, Richard E. Woods, Steve Eddins, 'Digital Image Processing using MATLAB', Pearson Education, Inc., 2004.

Paper Code: DSE15

Paper Title: Machine Learning (6 Credits, L+T+P = 4+0+2)

The objective of this course is to understand the basic theory underlying machine learning, to formulate machine learning problems corresponding to different applications. The students will understand a range of machine learning algorithms along with their strengths and weaknesses. To apply the algorithms to a real-world problem, optimize the models learned and report on the expected accuracy that can be achieved by applying the models.

On successful completion of the course, students will be able to:

1. Have a good understanding of the fundamental issues and challenges of machine learning: data, model selection, model complexity, etc.
2. Have an understanding of the strengths and weaknesses of many popular machine learning approaches.
3. Appreciate the underlying mathematical relationships within and across Machine Learning algorithms and the paradigms of supervised and un-supervised learning.
4. Be able to design and implement various machine learning algorithms in a range of real-world applications.

Unit	Content	No. of Lectures
------	---------	-----------------

1	Introduction: Concept of Machine Learning, Applications of Machine Learning, Key elements of Machine Learning, Supervised vs. Unsupervised Learning, Statistical Learning: Bayesian Method, The Naive Bayes Classifier	10
2	Softwares for Machine Learning and Linear Algebra Overview: Vectorization, Matrices and Vectors: Addition, Multiplication, Transpose and Inverse using available tools such as MATLAB.	10
3	Linear Regression: Prediction using Linear Regression, Gradient Descent, Linear Regression with one variable, Linear Regression with multiple variables, Polynomial Regression, Feature Scaling/Selection.	10
4	Logistic Regression: Classification using Logistic Regression, Logistic Regression vs. Linear Regression, Logistic Regression with one variable and with multiple variables	10
5	Regularization: Regularization and its utility: The problem of Overfitting, Application of Regularization in Linear and Logistic Regression, Regularization and Bias/Variance	10
6	Neural Networks: Introduction, Model Representation, Gradient Descent vs. Perceptron Training, Stochastic Gradient Descent, Multilayer Perceptrons, Multiclass Representation, Backpropagation Algorithm.	10
	<p>Practical:</p> <ol style="list-style-type: none"> 1. Perform elementary mathematical operations in Octave/MATLAB like addition, multiplication, division and exponentiation. 2. Perform elementary logical operations in Octave/MATLAB (like OR, AND, Checking for Equality, NOT, XOR). 3. Create, initialize and display simple variables and simple strings and use simple formatting for variables. 4. Create/Define single dimension / multidimensional arrays, and arrays with specific values like array of all ones, all zeros, array with random values within a range, or a diagonal matrix. 5. Use commands to compute the size of a matrix, size/length of a particular row/column, load data from a text file, store matrix data to a text file, finding out variables and their features in the current scope. 6. Perform basic operations on matrices (like addition, subtraction, multiplication) and display specific rows or columns of the matrix. 7. Perform other matrix operations like converting matrix data to absolute values, taking the negative of matrix values, adding/removing rows/columns from a matrix, finding the maximum or minimum values in a matrix or in a row/column, and finding the sum of some/all elements in a matrix. 8. Create various types of plots/charts like histograms, plot based on sine/cosine function based on data from a matrix. Further label different axes in a plot and data in a plot. 9. Generate different subplots from a given plot and color plot data. 10. Use conditional statements and different types of loops based on simple example/s. 11. Perform vectorized implementation of simple matrix operation like finding the transpose of a matrix, adding, subtracting or multiplying two matrices. 12. Implement Linear Regression problem. For example, based on a dataset consisting of the existing set of prices and area/size of the houses, predict the estimated price of a given house. 13. Based on multiple features/variables perform Linear Regression. For example, based on 	2 Credits

	<p>a number of additional features like number of bedrooms, servant room, number of balconies, number of houses of years a house has been built – predict the price of a house.</p> <p>14. Implement a classification/ logistic regression problem. For example based on different features of students data, classify, whether a student is suitable for a particular activity. Based on the available dataset, a student can also implement another classification problem like checking whether an email is spam or not.</p> <p>15. Use some function for regularization of the dataset based on problem 14.</p> <p>16. Use some function for neural networks, like Stochastic Gradient Descent or backpropagation - algorithm to predict the value of a variable based on the dataset of problem 14.</p>	
--	--	--

Reading list:

- Ethem Alpaydin, "Introduction to Machine Learning" 2nd Edition, The MIT Press, 2009.
- Tom M. Mitchell, "Machine Learning", First Edition by Tata McGraw-Hill Education, 2013.
- Christopher M. Bishop, "Pattern Recognition and Machine Learning" by Springer, 2007.
- Mevin P. Murphy, "Machine Learning: A Probabilistic Perspective" by The MIT Press, 2012.

Paper Code: DSE16

Paper Title: Introduction to Data Science (6 Credits, L+T+P = 4+0+2)

The objective of this course is to introduce the rapidly developing field of data science, which deals with the dramatic growth in the scale and complexity of data that can be collected and analysed. The course covers techniques for data analysis and decision making which requires interdisciplinary research in many areas, including algorithmic reasoning, statistics, technology competence, data handling, big data analysis / modelling, pattern recognition / artificial intelligence, enterprise / cloud computing, data visualisation / communication, as well as privacy and security. This course will introduce you to the core competencies of data science, starting from the philosophy and ethics of data collection and analysis, to data handling, to statistical modelling and machine learning, to scientific data visualisation and communication.

On successful completion of the course, students will be able to:

1. Demonstrate a solid understanding of the principles of data science.
2. Demonstrate practical knowledge in data preparation.
3. Use appropriate modelling and analyse techniques for data science problems.
4. Demonstrate competent skills in using data science technology.

Unit	Content	No. of Lectures																				
1	Data Scientist’s ToolBox : Turning data into actionable knowledge, introduction to the tools that will be used in building data analysis software: version control, markdown, git, GitHub, R, and RStudio.	10																				
2	R Programming Basics: Plotting of Data, Vectorization, Matrices and Vectors: Addition, Multiplication, Transpose and Inverse using available tools such as MATLAB.	10																				
3	Linear Getting and Cleaning Data: Obtaining data from the web, from APIs, from databases and from colleagues in various formats. Basics of data cleaning and making data —tidyl.	12																				
4	Exploratory Data Analysis: Essential exploratory techniques for summarizing data, applied before formal modeling commences, eliminating or sharpening potential hypotheses about the world that can be addressed by the data, common multivariate statistical techniques used to visualize high-dimensional data.	14																				
5	Reproducible Research: Concepts and tools behind reporting modern data analyses in a reproducible manner, To write a document using R markdown, integrate live R code into a literate statistical program, compile R markdown documents using knitr and related tools, and organize a data analysis so that it is reproducible and accessible to others.	14																				
	<p>Practical:</p> <ol style="list-style-type: none"> 1. Write a program that prints ‘Hello World’ to the screen. 2. Write a program that asks the user for a number n and prints the sum of the numbers 1 to n 3. Write a program that prints a multiplication table for numbers up to 12. 4. Write a function that returns the largest element in a list. 5. Write a function that computes the running total of a list. 6. Write a function that tests whether a string is a palindrome. 7. Implement linear search. 8. Implement binary search. 9. Implement matrices addition, subtraction and Multiplication 10. Fifteen students were enrolled in a course. Their ages were: 20 20 20 20 20 21 21 21 22 22 22 22 23 23 23 <ol style="list-style-type: none"> A. Find the median age of all students under 22 years B. Find the median age of all students C. Find the mean age of all students D. Find the modal age for all students E. Two more students enter the class. The age of both students is 23. What is now mean, mode and median? 11. Following table gives a frequency distribution of systolic blood pressure. Compute all the measures of dispersion. <table border="1" data-bbox="219 1661 1339 1787"> <tbody> <tr> <td>Midpoint</td> <td>95.5</td> <td>105.5</td> <td>115.5</td> <td>125.5</td> <td>135.5</td> <td>145.5</td> <td>155.5</td> <td>165.5</td> <td>175.5</td> </tr> <tr> <td>Number</td> <td>5</td> <td>8</td> <td>22</td> <td>27</td> <td>17</td> <td>9</td> <td>5</td> <td>5</td> <td>2</td> </tr> </tbody> </table> 12. Obtain probability distribution of X, where X is the number of spots showing when a six-sided symmetric die (i.e. all six faces of the die are equally likely) is rolled. Simulate 	Midpoint	95.5	105.5	115.5	125.5	135.5	145.5	155.5	165.5	175.5	Number	5	8	22	27	17	9	5	5	2	2 Credits
Midpoint	95.5	105.5	115.5	125.5	135.5	145.5	155.5	165.5	175.5													
Number	5	8	22	27	17	9	5	5	2													

	<p>random samples of sizes 40, 70 and 100 respectively and verify the frequency interpretation of probability.</p> <p>13. Make visual representations of data using the base, lattice, and ggplot2 plotting systems in R, apply basic principles of data graphics to create rich analytic graphics from available datasets.</p> <p>14. Use Git / Github software to create Github accounts. Also, create a repo using Github.</p>	
--	---	--

Reading list:

- Rachel Schutt, Cathy O'Neil, "Doing Data Science: Straight Talk from the Frontline" by Schroff/O'Reilly, 2013.
- Foster Provost, Tom Fawcett, "Data Science for Business" What You Need to Know About Data Mining and Data-Analytic Thinking" by O'Reilly, 2013.
- John W. Foreman, "Data Smart: Using data Science to Transform Information into Insight" by John Wiley & Sons, 2013.
- Ian Ayres, "Super Crunchers: Why Thinking-by-Numbers Is the New Way to Be Smart" 1st Edition by Bantam, 2007.
- Eric Seigel, "Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die", 1st Edition, by Wiley, 2013.
- Matthew A. Russel, "Mining the Social Web: Data mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More", Second Edition, by O'Reilly Media, 2013.

Paper Code: DSE17

Paper Title: Cloud Computing (6 Credits, L+T+P = 4+0+2)

The objective of this course is to provide the comprehensive and in-depth knowledge of Cloud Computing concepts, technologies, architecture and applications by introducing and researching state-of-the-art in Cloud Computing fundamental issues, technologies, applications and implementations. Another objective is to expose the students to frontier areas of Cloud Computing and information systems, while providing sufficient foundations to enable further study and research.

On successful completion of the course, students will be able to:

1. Articulate the main concepts, key technologies, strengths, and limitations of cloud computing and the possible applications of cloud computing.
2. Identify the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, etc.
3. Explain the core issues of cloud computing such as security, privacy, and interoperability.
4. Choose the appropriate technologies, algorithms, and approaches for the related issues.

- Identify problems, and explain, analyze, and evaluate various cloud computing solutions.

Unit	Content	No. of Lectures
1	Overview of Computing Paradigm: Recent trends in Computing: Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing.	8
2	Introduction to Cloud Computing: Introduction to Cloud Computing, History of Cloud Computing, Cloud service providers, Benefits and limitations of Cloud Computing.	7
3	Cloud Computing Architecture: Comparison with traditional computing architecture (client/server), Services provided at various levels, Service Models- Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service(SaaS), How Cloud Computing Works, Deployment Models- Public cloud, Private cloud, Hybrid cloud, Community cloud, Case study of NIST architecture.	20
4	Case Studies: Case study of Service model using Google App Engine, Microsoft Azure, Amazon EC2, Eucalyptus.	13
5	Service Management in Cloud Computing: Service Level Agreements (SLAs), Billing & Accounting, Comparing Scaling Hardware: Traditional vs. Cloud, Economics of scaling.	7
6	Cloud Security: Infrastructure Security- Network level security, Host level security, Application level security, Data security and Storage- Data privacy and security Issues, Jurisdictional issues raised by Data location, Authentication in cloud computing.	5
	<p>Practical:</p> <ol style="list-style-type: none"> Create virtual machines that access different programs on the same platform. Create virtual machines that access different programs on different platforms. Working on tools used in cloud computing online- <ol style="list-style-type: none"> Storage Sharing of data Manage your calendar, to-do lists, A document editing tool Exploring Google cloud Exploring microsoft cloud Exploring amazon cloud 	2 Credits

Reading list:

- Cloud Computing Bible, Barrie Sosinsky, Wiley-India, 2010
- Cloud Computing: Principles and Paradigms, Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wile, 2011
- Cloud Computing: Principles, Systems and Applications, Editors: Nikos Antonopoulos, Lee Gillam, Springer, 2012

- Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Ronald L. Krutz, Russell Dean Vines, Wiley-India, 2010
- Gautam Shroff, Enterprise Cloud Computing Technology Architecture Applications, Adobe Reader ebooks available from eBooks.com,2010
- Toby Velte, Anthony Velte, Robert Elsenpeter, Cloud Computing, A Practical Approach, McGraw Hills, 2010.
- Dimitris N. Chorafas, Cloud Computing Strategies, CRC Press, 2010.

Paper Code: DSE18

Paper Title: Numerical Methods (6 Credits, L+T+P = 4+0+2)

This course will help to grasp the basic elements of numerical methods and error analysis, understand the basics of approximation, integration and differentiation. To apply the numerical solution of differential equations and to solve linear & nonlinear algebraic equations.

On successful completion of the course, students will be able to:

1. Understand common numerical methods and how they are used to obtain approximate solutions to otherwise intractable mathematical problems.
2. Apply numerical methods to obtain approximate solutions to mathematical problems.
3. Derive numerical methods for various mathematical operations and tasks, such as interpolation, differentiation, integration, the solution of linear and nonlinear equations, and the solution of differential equations.
4. Analyse and evaluate the accuracy of common numerical methods.
5. Implement numerical methods in various programming languages.

Unit	Content	No. of Lectures
1	Floating point representation and computer arithmetic, Significant digits, Errors: Round-off error, Local truncation error, Global truncation error, Order of a method, Convergence and terminal conditions, Efficient computations	8
2	Bisection method, Secant method, Regula-Falsi method Newton-Raphson method, Newton's method for solving nonlinear systems Gauss elimination method (with row pivoting) and Gauss-Jordan method, Gauss Thomas method for tridiagonal systems	7

3	Iterative methods: Jacobi and Gauss-Seidel iterative methods Interpolation: Lagrange's form and Newton's form Finite difference operators, Gregory Newton forward and backward differences Interpolation Piecewise polynomial interpolation: Linear interpolation, Cubic spline interpolation (only method), Numerical differentiation: First derivatives and second order derivatives, Richardson extrapolation	20
4	Numerical integration: Trapezoid rule, Simpson's rule (only method), Newton-Cotes open formulas Extrapolation methods: Romberg integration, Gaussian quadrature, Ordinary differential equation: Euler's method	13
5	Modified Euler's methods: Heun method and Mid-point method, Runge-Kutta second methods: Heun method without iteration, Mid-point method and Ralston's method Classical 4th order Runge- Kutta method, Finite difference method for linear ODE	7
	<p>Practical:</p> <ol style="list-style-type: none"> 1. Find the roots of the equation by bisection method. 2. Find the roots of the equation by secant/Regula-Falsi method. 3. Find the roots of the equation by Newton's method. 4. Find the solution of a system of nonlinear equations using Newton's method. 5. Find the solution of the tridiagonal system using Gauss Thomas method. 6. Find the solution of the system of equations using the Jacobi/Gauss-Seidel method. 7. Find the cubic spline interpolating function. 8. Evaluate the approximate value of finite integrals using Gaussian/Romberg integration. 9. Solve the boundary value problem using a finite difference method. <p>Note: Programming is to be done in any one of Computer Algebra Systems: MATLAB / MATHEMATICA / MAPLE.</p>	2 Credits

Reading list:

- Laurence V. Fausett, Applied Numerical Analysis, Using MATLAB, Pearson, 2/e (2012)
- M.K. Jain, S.R.K. Iyengar and R.K. Jain, Numerical Methods for Scientific and Engineering Computation, New Age International Publisher, 6/e (2012)
- Steven C Chapra, Applied Numerical Methods with MATLAB for Engineers and Scientists, Tata McGraw Hill, 2/e (2010)

Paper Code: DSE19

Paper Title: Systems Programming (6 Credits, L+T+P = 4+0+2)

The objective of this course is to introduce students to the fundamental model of the processing of high level language programs for execution on computer systems. To understand and implement Assembler, Loader, Linkers, Macros & Compilers.

On successful completion of the course, students will be able to:

1. To understand the basics of system programs like editors, compiler, assembler, linker, loader, interpreter and debugger.

2. Understand the task of translation of assembly language and high-level language programs into machine language programs.
3. Implement symbols tables.
4. The task of code optimization.
5. The various types of linking of program modules. To understand how linker and loader create an executable program from an object module created by assembler and compiler.

Unit	Content	No. of Lectures
1	Assemblers & Loaders, Linkers: One pass and two pass assembler, design of an assembler, Absolute loader, relocation and linking concepts, relocating loader and Dynamic Linking.	10
2	Introduction: Overview of compilation, Phases of a compiler.	5
3	Lexical Analysis: Role of a Lexical analyzer, Specification and recognition of tokens, Symbol table, lex.	6
4	Parsing: Bottom up parsing- LR parser, yacc.	10
5	Intermediate representations: Three address code generation, syntax directed translation, translation of types, control statements.	10
6	Storage organization: Activation records, stack allocation	9
7	Code Generation: Object code generation	10
	Practical: <ol style="list-style-type: none"> 1. To implement an assembler for a hypothetical language. 2. To get familiar with lex: write a program to recognize numbers, identifiers. 3. To get familiar with yacc: write a desk calculator 	2 Credits

Reading list:

- Santanu Chattopadhyaya, Systems Programming, PHI, 2011.
- Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Compilers: Principles, Techniques, and Tools, 2nd edition, Prentice Hall, 2006.
- D. M. Dhamdhare, Systems Programming, Tata McGraw Hill, 2011.
- Leland Beck, D. Manjula, System Software: An Introduction to System Programming, 3rd edition, Pearson Education, 2008.

- Grune D, Van Reeuwijk . K, Bal H. E, Jacobs C J H, Langendoen K, Modern Compiler Design, 2nd edition, Springer, 2012

Paper Code: DSE20

Paper Title: Operational Research (6 Credits L+T+P = 4+0+2)

The objective of the course is to impart knowledge in concepts and tools of operational research, understand mathematical models used in operational research and apply these techniques constructively to make effective business decisions.

On successful completion of the course, students will be able to:

1. Formulate and obtain the optimal solution for Linear Programming problems.
2. Plan, schedule and control the given project.
3. Understand the need of inventory management.
4. Choose the appropriate queuing model for a given practical application.

Unit	Content	No. of Lectures
1	Introduction to Operational Research (OR): Origin & Development, Different Phases of OR study, Methodology of OR, Scope and Limitations of OR, Applications of OR	10
2	Linear Programming: Linearly independent / dependent vectors, Basis, Convex sets, Extreme points. Graphical method. Simplex method, Artificial variable techniques- Two Phase Method; M-Charnes Method, Special cases in LPP.	2
3	Duality: Definition of the dual problem, Primal-dual relationships, Economic Interpretation of Duality, Dual simplex Method.	6
4	Sensitivity analysis: Changes in cost and resource vector	10
	<p>Practical:</p> <ol style="list-style-type: none"> 1. To solve Linear Programming Problem using Graphical <ol style="list-style-type: none"> a. Method with Unbounded solution b. Infeasible solution c. Alternative or multiple solutions. 2. Solution of LPP with simplex method. 3. Problem solving using M-Charnes method. 4. Problem solving using Two Phase method. 5. Illustration of following special cases in LPP using Simplex method <ol style="list-style-type: none"> a. Unrestricted variables b. Unbounded solution c. Infeasible solution\ d. Alternative or multiple solution 6. Problems based on Dual simplex method. 	2 Credits

	7. Problems based on sensitivity analysis.	
--	--	--

Reading list:

- G. Hadley: Linear Programming. Narosa, 2002 (reprint).
- A. Ravindran, D. T. Phillips and James J. Solberg: Operations Research-Principles and Practice, John Wiley & Sons, 2005.
- Hamdy A. Taha: Operations Research-An Introduction, Prentice Hall, 8th Edition, 2008.
- F.S. Hillier. G.J. Lieberman: Introduction to Operations Research- Concepts and Cases, 9th Edition, Tata McGraw Hill. 2010.

Paper Code: DSE21

Paper Title: Data Mining (6 Credits L+T+P = 4+0+2)

This course will introduce students to the basic concepts and techniques of data mining, develop skills of using recent data mining software for solving practical problems, study the methodology of engineering legacy databases for data warehousing and data mining to derive business rules for decision support systems, develop and apply critical thinking, problem-solving, and decision-making skills.

On successful completion of the course, students will be able to:

1. Understand data mining principles and techniques and acquaint the students with the data mining techniques for building competitive advantage through proactive analysis, predictive modelling, and identifying new trends and behaviors.
2. Learn how to gather and analyze large sets of data to gain useful business understanding.
3. Produce a quantitative analysis report/memo with the necessary information to make decisions.
4. Describing and demonstrating basic data mining algorithms, methods, and tools.
5. Identifying business applications of data mining
6. Overview of the developing areas - web mining, text mining, and ethical aspects of data mining.

Unit	Content	No. of
------	---------	--------

		Lectures
1	Overview: Predictive and descriptive data mining techniques, supervised and unsupervised learning techniques, process of knowledge discovery in databases, pre-processing methods	15
2	Data Mining Techniques: Association Rule Mining, classification and regression techniques, clustering, Scalability and data management issues in data mining algorithms, measures of interestingness	20
3	Classification: Binary Classification - Basic concepts, Bayes theorem and Naive Bayes classifier, Association based classification, Rule based classifiers, Nearest neighbour classifiers, Decision Trees, Random Forest; Perceptrons; Multi-category classification; Model overfitting, Evaluation of classifier performance - cross validation, ROC curves	20
4	Applications: Text mining, Web data analysis, Recommender systems.	5
	Practical: 1. Practical exercises based on concepts listed in theory.	2 Credits

Reading list:

- Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Introduction to Data Mining, Pearson Education.2005.
- Richard Roiger, Michael Geatz, Data Mining: A Tutorial Based Primer, Pearson Education 2003.
- G.K. Gupta, Introduction to Data Mining with Case Studies, PHI, 2006.
- Soman K P, Diwakar Shyam, Ajay V Insight Into Data Mining: Theory And Practice, PHI, 2006

Paper Code: DSE22

Paper Title: Project work/ Dissertation (6 Credits, L+T+P = 4+0+2)

1. This option will be offered only in the 6th Semester.
2. The students will be allowed to work on any project based on the concepts studied in core / elective or skill based elective courses.
3. The group size should be a maximum of three (03) students.
4. Each group will be assigned a teacher as a supervisor who will handle both their theory as well lab classes.
5. A maximum of Four (04) projects would be assigned to one teacher.

Paper Code: DSE23

Paper Title: Big Data Analytics (6 Credits, L+T+P = 4+0+2)

This course will teach you to understand the Big Data Platform and its use cases, provide an overview of Apache Hadoop, provide HDFS Concepts and Interfacing with HDFS, understand Map Reduce Jobs, provide hands on Hadoop Ecosystem, apply analytics on Structured, Unstructured Data.

On successful completion of the course, students will be able to:

1. Identify Big Data and its Business Implications.
2. List the components of Hadoop and Hadoop Ecosystem.
3. Access and Process Data on Distributed File System.
4. Manage Job Execution in Hadoop Environment.
5. Develop Big Data Solutions using Hadoop Ecosystem.

Unit	Content	No. of Lectures
1	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.	10
2	Introduction to Hadoop and Hadoop Architecture: Big Data Apache Hadoop and Hadoop EcoSystem, Moving Data in and out of Hadoop, Understanding inputs and outputs of MapReduce, Data Serialization	10
3	HDFS, HIVE and HIVEQL, HBAS: HDFS Overview, Installation and Shell, Java API; Hive Architecture and Installation, Comparison with Traditional Database, HiveQL Querying Data, Sorting and Aggregating, Map Reduce Scripts, Joins & Sub queries, 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.	10
4	SPARK: Introduction to Data Anaysis with Spark, Downloading Spark and getting started, Programming with RDDs, Machine Learning with Mlib.	10
5	HADOOP ECOSYSTEM and YARN: Hadoop ecosystem components-Schedulers – Fair and capacity, Hadoop 2.0 New Features- NameNode High Availability, HDFS Federation, MRv2, YARN, Running MRv1 in YARN	10
6	Database for the Modern Web: Introduction to MongoDB key features, Core Server tools, MongoDB through the JavaScript's Shell, Creating and Querying through Indexes, Document Oriented, principles of schema design, Constructing queries on Databases, collections and Documents, MongoDB Query Language	10

	Practical: 1. Practical exercises based on concepts listed in theory.	2 Credits
--	---	-----------

Reading list:

- BIG Data and Analytics , Sima Acharya, Subhashini Chhellappan, Willey
- MongoDB in Action, Kyle Banker, Piter Bakkum , Shaun Verch, Dream tech Press
- Tom White, “HADOOP: The definitive Guide”, O Reilly 2012.
- VigneshPrajapati, “Big Data Analytics with R and Haoop”, Packt Publishing 2013

Paper Code: DSE24

Paper Title: Soft Computing (6 Credits, L+T+P = 4+0+2)

This course will teach you to understand soft computing concepts, technologies, and applications, understand the underlying principle of soft computing with its usage in various applications and understand different soft computing tools to solve real life problems.

On successful completion of the course, students will be able to:

1. Learn fuzzy logic and its applications.
2. Learn artificial neural networks and its applications.
3. Solving single-objective optimization problems using Genetic Algorithms.
4. Apply of soft computing to solve problems in varieties of application domains.

Unit	Content	No. of Lectures
1	Introduction to Artificial Intelligence System, Neural Network, Fuzzy Logic & Genetic Algorithm. Fuzzy Set Theory : Fuzzy Versus Crisp, Crisp Set, Fuzzy Set, Crip Relation, Fuzzy Relations.	10
2	Fuzzy System: Crisp Logic, Predicate Logic, Fuzzy Logic, Fuzzy Rule Based System, Defuzzification Methods, Applications.	5
3	Genetic Algorithms, Basic Concepts, Creation Of Offspring, Working Principle, Encoding, Fitness Function, Reproduction.	5
4	Genetic Modelling, Inheritance Operations, Cross Over, Inversion And Deletion, Mutation Operator, Bit Wise Operators, Generation Cycle, Convergence Of Genetic Algorithm, Application, Multi Level Optimization, Real Life Problems, Difference And Similarities Between GA And Other Traditional Methods, Advanced In GA.	10
5	Fundamentals Of Neural Networks, Basic Concepts Of Neural Network, Human Brain, Model	10

	Of An Artificial Neuron, Neural Network Architectures, Characteristic Of Neural Networks, Learning Method, Taxonomy Of Neural Network Architectures, History Of Neural Network Research, Early Neural Network Architectures, Some Application Domains.	
6	Back Propagation Network Architecture Of Back Propagation Network, Back Propagation Learning, Illustration, Applications, Effect Of Tuning Parameters Of The Back Propagation Neural Network, Selection Of Various Parameters In BPN, Variations Of Standard Back Propagation Algorithm.	10
7	Associative Memory And Adaptive Resonance Theory, Autocorrelation, Hetrocorrelators , Multiple Training Encoding Strategy, Exponential BAM, Associative Memory For Real Coded Pattern Pairs, Applications, Introduction To Adaptive Resonance Theory, ARTI, Character Recognition Using ARII	10
	Practical: 1. Practical exercises based on concepts listed in theory.	2 Credits

Reading list:

- S. Rajasekharan & G. A. Vijayalakshmi – “ Neural Network, Fuzzy Logic And Genetic Algorithm Synthesis And Applications”, Prentice Hall Of India PLT, Pai – 2004
- Jyh – Shing R Jang, C. T. Sun, E Mizutani – Neuro Fuzzy And Soft Computing – A Computational Approach To Learning And Machine Intelligence”, Prentice Hall Of India – 1997.

Skill Enhancement Courses

Paper Code: SEC01

Paper Title: Android Programming (2 Credits, L+T+P = 1+0+1)

This course is intended to give the students an introduction to the Android operating system, its architecture, and development tools. It also will give an exposure to databases, object-oriented programming and interface designing.

On successful completion of the course, students will be able to:

1. Understand the basic android architecture.
2. Use Android development tools.
3. Solve programming problems using object-oriented features.
4. Design interfaces.
5. Connect with databases.

Unit	Content	No. of Lectures
------	---------	-----------------

1	Introduction: History of Android, Introduction to Android Operating Systems, Android Development Tools, Android Architecture.	2
2	Overview of object oriented programming using Java: OOPs Concepts: Inheritance, Polymorphism, Interfaces, Abstract class, Threads, Overloading and Overriding, Java Virtual Machine.	4
3	Development Tools: Installing and using Eclipse with ADT plug-in, Installing Virtual machine for Android sandwich/Jelly bean (Emulator), configuring the installed tools, creating an android project - Hello Word, run on emulator, Deploy it on USB-connected Android device.	4
4	User Interface Architecture: Application context, intents, Activity life cycle, multiple screen sizes	1
5	User Interface Design: Form widgets, Text Fields, Layouts, Button control, toggle buttons, Spinners (Combo boxes), Images, Menu, Dialog.	2
6	Database: Understanding of SQLite database, connecting with the database.	2
	<p>Practical:</p> <ol style="list-style-type: none"> 1. Create - Hello World application. That will display- Hello World in the middle of the screen in the emulator. Also display ``Hello World`` in the middle of the screen in the Android Phone. 2. Create an application with a login module. (Check username and password). 3. Create a spinner with strings taken from a resource folder (res >> value folder) and on changing the spinner value, Image will change. 4. Create a menu with 5 options and the selected option should appear in the text box. 5. Create a list of all courses in your college and on selecting a particular course teacher in-charge of that course should appear at the bottom of the screen. 6. Create an application with three option buttons, on selecting a button colour of the screen will change. 7. Create and Login application as above. On successful login, pop up the message. 8. Create an application to Create, Insert, update, Delete and retrieve operation on the database. 	1 Credit

Reading list:

1. Android application development for java programmers. By James C. Sheusi. Publisher: Cengage Learning, 2013.
2. <http://www.developer.android.com>
3. <http://developer.android.com/about/versions/index.html>
4. <http://developer.android.com/training/basics/firstapp/index.html>
5. <http://docs.oracle.com/javase/tutorial/index.htm> (Available in the form of free downloadable ebooks also).
6. <http://developer.android.com/guide/components/activities.html>
7. <http://developer.android.com/guide/components/fundamentals.html>
8. <http://developer.android.com/guide/components/intents-filters.html>
9. <http://developer.android.com/training/multiscreen/screensizes.html>

10. <http://developer.android.com/guide/topics/ui/controls.html>
11. <http://developer.android.com/guide/topics/ui/declaring-layout.html>
12. <http://developer.android.com/training/basics/data-storage/databases.html>

Paper Code: SEC02

Paper Title: Programming in MATLAB (2 Credits, L+T+P = 1+0+1)

This course is intended to give the students an introduction to the MATLAB programming language. It also will give an exposure to different programming constructs like control statements, procedures, functions, graphs and interfaces.

On successful completion of the course, students will be able to:

1. Understand basic programming concepts.
2. Use various programming constructs of MATLAB to solve problems.
3. Plot graphs.
4. Design interfaces.

Unit	Content	No. of Lectures
1	Introduction to Programming: Components of a computer, working with numbers, Machine code, Software hierarchy.	2
2	Programming Environment: MATLAB Windows, A First Program, Expressions, constants, Variables and assignment statement, Arrays.	3
3	Graph Plots: Basic plotting, Built in functions, generating waveforms, Sound, replay, load and save.	2
4	Procedures and Functions: Arguments and return values, M-files, Formatted console inputOutput, String handling.	3
5	Control Statements: Conditional statements: If, Else, Else-if, Repetition statements, While, For loop	2
6	Manipulating Text: Writing to a text file, Reading from a text file, Randomising and sorting A list, searching a list.	2
7	GUI Interface: Attaching buttons to actions, Getting Input, Setting Output.	2
	Practical:	1 Credit

1. Write a program to assign the following expressions to a variable A and then to print out the value of A.
 - a. $(3+4)/(5+6)$
 - b. $2\pi^2$
 - c. $\sqrt{2}$
 - d. $(0.0000123 + 5.67 \times 10^{-3}) \times 0.4567 \times 10^{-4}$
2. Celsius temperatures can be converted to Fahrenheit by multiplying by 9, dividing by 5, and adding 32. Assign a variable called C the value 37, and implement this formula to assign a variable F the Fahrenheit equivalent of 37 Celsius.
3. Set up a vector called N with five elements having the values: 1, 2, 3, 4, 5. Using N, create assignment statements for a vector X which will result in X having these values:
 - a. 1/2, 1, 3/2, 2, 5/2
 - b. 2, 4, 6, 8, 10
 - c. 1, 1/2, 1/3, 1/4, 1/5
 - d. 1, 1/4, 1/9, 1/16, 1/25
4. A supermarket conveyor belt holds an array of groceries. The price of each product (in pounds) is [0.6, 1.2, 0.5, 1.3]; while the numbers of each product are [3, 2, 1, 5]. Use MATLAB to calculate the total bill.
5. The sortrows(x) function will sort a vector or matrix X into increasing row order. Use this function to sort a list of names into alphabetical order.
6. The identity matrix is a square matrix that has ones on the diagonal and zeros elsewhere. You can generate one with the eye() function in MATLAB. Use MATLAB to find a matrix B, such that when multiplied by matrix A=[1 2; -1 0] the identity matrix I=[1 0; 0 1] is generated. That is A*B=I.
7. **Create an array of N numbers. Now find a single MATLAB statement that picks out from that array the 1,4,9,16,...,Nth entries, i.e. those numbers which have indices that are square numbers.**
8. Draw a graph that joins the points (0,1), (4,3), (2,0) and (5,-2).
9. The seeds on a sunflower are distributed according to the formula below. Plot a small circle at each of the first 1000 coordinates :

$$r_n = \sqrt{n}$$

$$\theta_n = \frac{137.51}{180} \pi n$$
10. Calculate 10 approximate points from the function $y=2x$ by using the formulae:
 - a. $x_n = n$
 - b. $y_n = 2n + \text{rand} - 0.5$
 Fit a line of best fit to these points using the function polyfit() with degree=1, and generate coordinates from the line of best fit using polyval(). Use the on-line help to find out how to use these functions. Plot the raw data and the line of best fit.
11. Calculate and replay 1 second of a sine wave at 500 Hz with a sampling rate of 11025 Hz. Save the sound to a file called "ex35.wav". Plot the first 100 samples.
12. Calculate and replay a 2 second chirp. That is, a sinusoid that steadily increases in frequency with time, from say 250Hz at the start to 1000Hz at the end.
13. Build a square wave by adding together 10 odd harmonics: 1f, 3f, 5f, etc. The amplitude of the nth harmonic should be 1/n. Display a graph of one cycle of the result superimposed on the individual harmonics.

	<p>14. Write a function called FtoC (ftoc.m) to convert Fahrenheit temperatures into Celsius. Make sure the program has a title comment and a help page. Test from the command window with:</p> <ol style="list-style-type: none"> a. FtoC(96) b. Look for Fahrenheit c. help FtoC <p>15. Write a program to input 2 strings from the user and to print out (i) the concatenation of the two strings with a space between them, (ii) a line of asterisks the same length as the concatenated strings, and (iii) the reversed concatenation. For example:</p> <ol style="list-style-type: none"> a. Enter string 1: <u>Mark</u> b. Enter string 2: <u>Huckvale</u> c. Mark Huckvale d. elavkcuH kraM 	
--	--	--

Reading list:

1. MATLAB: An Introduction with Applications, by Amos Gilat, 2nd edition, Wiley, 2004
2. C.B. Moler, Numerical Computing with MATLAB, SIAM, 2004.

Paper Code: SEC03

Paper Title: HTML Programming (2 Credits L+T+P = 1+0+1)

This course is intended to give the students an introduction to the Hypertext Markup Language and its various components. Use of different HTML components like links, images, tables, headers, forms, CSS, etc., to design an ergonomic and efficient webpage will be taught in this course.

On successful completion of the course, students will be able to:

1. Understand basics of HTML.
2. Use various components of HTML to design a web page.
3. Style with CSS.

Unit	Content	No. of Lectures
1	Introduction	1
2	The Basics: The Head, the Body, Colors, Attributes, Lists, ordered and unordered	2

3	Links: Introduction, Relative Links, Absolute Links, Link Attributes, Using the ID Attribute to Link Within a Document	3
4	Images: Putting an Image on a Page, Using Images as Links, Putting an Image in the Background	2
5	Table: Creating a Table, Table Headers, Captions, Spanning Multiple columns, Styling Table	4
6	Form: Basic Input and Attributes, Other Kinds of Inputs, Styling forms with CSS	3

Practical:

1. Create an HTML document with the following formatting options:
 - a. Bold
 - b. Italics
 - c. Font (Type, Size and Color)
2. Create an HTML document which consists of:
 - a. Ordered List
 - b. Unordered List
 - c. Nested List
 - d. Image
3. Create an HTML document which implements Internal linking as well as External linking.
4. Create a table using HTML which consists of columns for Roll No., Student's name and grade.

Result		
Roll No	Name	Grade

5. Create a Table with the following view:

1 Credit

--	--

6. Create a form using HTML which has the following types of controls:
- Text box
 - Option/radio buttons
 - Check boxes
 - Reset and Submit buttons

Subscribe to XYZ News Magazine and Emails

Interested in receiving daily small updates of all latest News? Well, now you can. And best of all, it is free! Just fill out this form and submit it by clicking the "send it In" button. We will put you on our mailing list and you will receive your first email in 3-5 days.

Please fill the following boxes to help us send the emails and our news letter:

First Name:

Last Name:

Business:

We must have a correct e-mail address to send you the news letter.

Email:

How did you hear about XYZ News Magazine and Emails?

Here on the Web In a magazine Television Other

Would you like to be on our regular mailing list?

Yes, we love junk emails

7. Create HTML documents (having multiple frames) in the following three formats:

Frame 1	
Frame 2	
Frame 1	
Frame 2	Frame 3

Reading list:

- Virginia DeBolt , Integrated HTML and CSS A Smarter, Faster Way to Learn Wiley / Sybex , 2006
- Cassidy Williams, Camryn Williams Introduction to HTML and CSS, O'Reilly, 2015

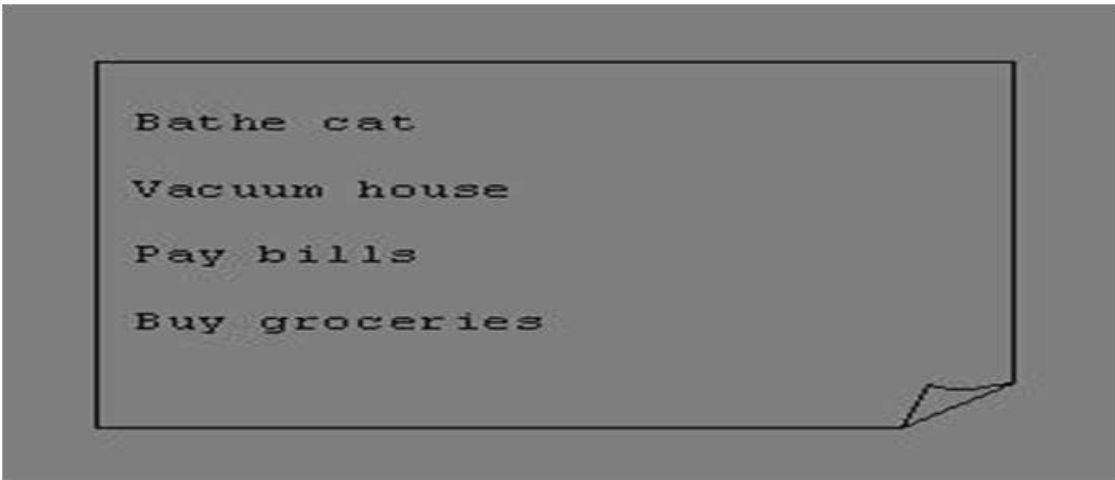
Paper Code: SEC04

Paper Title: XML Programming (2 Credits, L+T+P = 1+0+1)

This course is intended to give the students an introduction to the Extensible Markup Language and its various components. It will expose the students to XML concepts like structure , syntax, document classes, rules, scripting, linking, style sheets etc.

On successful completion of the course, students will be able to:

1. Understand basics of XML.
2. Understand the structure and syntax of XML.
3. Understand concepts like scripting, linking, etc.
4. Use style sheets.

Unit	Content	No. of Lectures
1	Introduction: Understanding Mark-up Languages, Introduction to XML and its Goals.	3
2	XML Basics: XML Structure and Syntax, Document classes and Rules.	5
3	XML Concepts: Scripting XML, XML as Data, Linking with XML.	4
4	XML with Style: XSL –Style Sheet Basics, XSL basics, XSL style sheets.	3
	<p>Practical:</p> <p>1. Information Structure: In this exercise, student will practice identifying the structure of an information object. For the sample document provided below: Label the information structures you see, including containing structures.</p> <p>a. Draw a tree representation of the structure.</p> 	1 Credit

2. Deconstructing an XML Document: In this exercise, students will practice identifying the explicit structure within an XML document. In a sense, this is the reverse of what you did in Exercise 1. For the sample XML markup below, create a document-like representation (or a simple drawing) for the content contained within the XML tags:

```
<book>
  <coverInfo>
    <title>The XML Handbook</title>
    <author>Charles F. Goldfarb</author>
    <author>Paul Prescod</author>
    <edition>Second</edition>
    <description>The definitive XML resource: applications, products, and
    technologies. Revised and expanded—over 600 new pages.
  </description>
  </coverInfo>
</book>
```

3. Creating XML Markup: In this exercise, create some XML markup based on the tree representation from Exercise #1 above, and the content from the original sample document.

4. Well-Formedness: This exercise checks your understanding of the constraints for well-formedness. Are the following document instances well-formed? Explain any NO answers.

```
<list><title>The first list</title><item>An
item</list> <item>An item</item><item>Another
item</item>
```

```
<para>Bathing a cat is a <emph>relatively</emph> easy task as long as the cat is
willing.</para> <bibl><title>How to Bathe a Cat<author></title>Merlin
Bauer<author></bibl>
```

5. Well Formedness: This exercise is a bit more challenging than the previous example. Here is a fragment of an XML document instance. Identify all the places where it fails to match the constraints for well formedness.

```
<PROCEDURE><TITLE>How to Bathe a Cat</TITLE> <OVERVIEW>
This procedure tells you how to bathe a cat. <WARNING></OVERVIEW>
Cats don't like to take baths. You could get hurt doing this. Be sure to obtain all the
required protective gear before you start.
</WARNING><EQUIPEMENT><ITEM>Hockey Mask <ITEM>Padded Full-
body Kevlar Armor</ITEM><ITEM>Tub full of
warmwater</ITEM><ITEM>Towels
</ITEM><ITEM>First Aid kit</ITEM><ITEM>Cat Shampoo</ITEM>
<EQUIPMENT><INSTRUCTIONS> <STEP> Locate the cat, who by now is hiding
under the bed.</STEP><STEP>Place the cat in the tub of water.</STEP>
<ITEM>Using the First Aid kit, repair the damage to your head and arms.</STEP>
<STEP>Place the cat back in the tub and hold it down.</STEP> <STEP>Wash it
really fast, then make an effort to dry it with the towels.</STEP> <STEP>Decide
not to do this again. </STEP> </INSTRUCTIONS>
```

Note: Cover more exercises based on XML Programming theory concepts.

Reading list:

- XML in action web technology by William J. Pardi
- Step by Step XML by Michael J. Young

Paper Code: SEC05**Paper Title: Oracle (SQL/PL-SQL) (2 Credits, L+T+P = 1+0+1)**

This course is intended to give the students an introduction to the Oracle database management system and enable them to use various PL/SQL commands to access the Oracle DBMS to effectively design complex databases for various use cases.

On successful completion of the course, students will be able to:

1. Understand basics of the Oracle DBMS.
2. Use various PL/SQL commands for tasks like managing tables and data in Oracle.
3. Perform crucial database functions like transaction commit, rollback, etc., in Oracle.

Unit	Content	No. of Lectures
1	SQL Vs. SQL * Plus: SQL Commands and Data types, Operators and Expressions, Introduction to SQL * Plus	2
2	Managing Tables and Data: Creating and Altering Tables (Including constraints), Data Manipulation Command like Insert, update, delete, SELECT statement with WHERE, GROUP BY and HAVING, ORDER BY, DISTINCT, Special operator like IN, ANY, ALL BETWEEN, EXISTS, LIKE, Join, Built in functions	4
3	Other Database Objects: View, Synonyms, Index	2
4	Transaction Control Statements: Commit, Rollback, Savepoint	2
5	Introduction to PL/SQL: SQL v/s PL/SQL, PL/SQL Block Structure, Language construct of PL/SQL (Variables, Basic and Composite Data type, Conditions looping etc.), % TYPE and % ROWTYPE	5

<p>Practical: [SQL COMMANDS]</p> <ol style="list-style-type: none"> 1. SQL* formatting commands 2. To create a table, alter and drop tables. 3. To perform select, update, insert and delete operations in a table. 4. To make use of different clauses viz where, group by, having, order by, union and intersection, 5. To study different constraints. <p>[SQL FUNCTION]</p> <ol style="list-style-type: none"> 6. To use oracle function viz aggregate, numeric, conversion, string function. 7. To understand use and working with joins. 8. To make use of transaction control statement viz rollback, commit and save point. 9. To make views of a table. 10. To make indexes of a table. <p>[PL/SQL]</p> <ol style="list-style-type: none"> 11. To understand working with PL/SQL 12. To implement Cursor on a table. 13. To implement trigger on a table 	<p>1 Credit</p>
---	-----------------

Reading list:

- Ivan Bayross, "SQL, PL/SQL the Programming Language of Oracle Paperback ", BPB Publications, 2010.
- Steven Feuerstein, Bill Pribyl , "Oracle PL/SQL Programming", 6th Edition, O'Reilly Media, 2014.
- Rajeeb C. Chatterjee, "Learning Oracle SQL and PL/SQL: A simplified Guide", PHI, 2012.
- Ron Hardman, Michael Mclaughlin, "Expert Oracle PL/SQL", Oracle Press, 2005.
- Michael Mclaughlin, "Oracle Database 11g PL/SQL Programming", Oracle Press, 2008.
- John Watson, Roopesh Ramklass, "OCA Oracle Database 11g SQL Fundamentals I Exam Guide", Oracle Press, 2008.

Paper Code: SEC06

Paper Title: Programming in Python (2 Credits, L+T+P = 1+0+1)

This course is intended to give the students an introduction to the Python Programming Language and its various components. It will expose the students to various programming solving techniques like flowchart, decision table, algorithms, etc.

On successful completion of the course, students will be able to:

1. Understand basics of Python programming language.
2. Use flowcharts, decision table, etc., to solve problems.
3. Use basic programming constructs like loops, control statements, etc., to solve problems

Unit	Content	No. of Lectures
1	Planning the Computer Program: Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation.	2
2	Techniques of Problem Solving: Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming.	2
3	Overview of Programming: Structure of a Python Program, Elements of Python	3
4	Introduction to Python: Python Interpreter, Using Python as calculator, Python shell, Indentation. Atoms, Identifiers and keywords, Literals, Strings, Operators (Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bit wise operator, Increment or Decrement operator.	4
5	Creating Python Programs: Input and Output Statements, Control statements (Branching, Looping, Conditional Statement, Exit function, Difference between break, continue and pass.), Defining Functions, default arguments.	4
	<p>Practical:</p> <p>Section: A (Simple Program)</p> <ol style="list-style-type: none"> 1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon users choice. 2. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria : <ul style="list-style-type: none"> Grade A: Percentage ≥ 80 Grade B: Percentage ≥ 70 and < 80 Grade C: Percentage ≥ 60 and < 70 Grade D: Percentage ≥ 40 and < 60 Grade E: Percentage < 40 3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from the user. 4. WAP to display the first n terms of Fibonacci series. 5. WAP to find the factorial of the given number. 6. WAP to find the sum of the following series for n terms: $1 - 2/2! + 3/3! - n/n!$ 7. WAP to calculate the sum and product of two compatible matrices. <p>Section: B (Visual Python):</p> <p><i>All the programs should be written using user defined functions, wherever possible.</i></p> <ol style="list-style-type: none"> 1. Write a menu-driven program to create mathematical 3D objects <ol style="list-style-type: none"> a. Curve b. Sphere c. Cone d. Arrow e. Ring f. cylinder. 	1 Credit

	<ol style="list-style-type: none"> 2. WAP to read n integers and display them as a histogram. 3. WAP to display sine, cosine, polynomial and exponential curves. 4. WAP to plot a graph of people with pulse rate p vs. height h. The values of p and h are to be entered by the user. 5. WAP to calculate the mass m in a chemical reaction. The mass m (in gms) disintegrates according to the formula $m=60/(t+2)$, where t is the time in hours. Sketch a graph for t vs. m, where $t \geq 0$. 6. A population of 1000 bacteria is introduced into a nutrient medium. The population p grows as follows: $P(t) = (15000(1+t))/(15+ e)$ where the time t is measured in hours. WAP to determine the size of the population at a given time t and plot a graph for P vs t for the specified time interval. 7. Input initial velocity and acceleration, and plot the following graphs depicting equations of motion: <ol style="list-style-type: none"> a. velocity wrt time ($v=u+at$) b. distance wrt time ($s=u*t+0.5*a*t*t$) c. distance wrt velocity ($s=(v*v-u*u)/2*a$) 8. WAP to show a ball bouncing between 2 walls. (Optional) 	
--	---	--

Reading list:

1. T. Budd, Exploring Python, TMH, 1st Ed, 2011
2. Python Tutorial/Documentation www.python.org 2015
3. Allen Downey, Jeffrey Elkner, Chris Meyers , How to think like a computer scientist: learning with Python , Freely available online.2012
4. <http://docs.python.org/3/tutorial/index.html>
5. <http://interactivepython.org/courselib/static/pythonds>
6. <http://www.ibiblio.org/g2swap/byteofpython/read/>

Paper Code: SEC07

Paper Title: UNIX/LINUX Programming (2 Credits, L+T+P = 1+0+1)

This course is intended to give the students an introduction to the Linux/Unix operating system. It will give the students a hands-on exposure to the various system level tasks that can be accomplished by using an OS. It will also introduce the students to shell programming and its various components..

On successful completion of the course, students will be able to:

1. Understand basics of an OS.
2. Use Linux/Unix commands to interact with the OS.
3. Use basic programming constructs like loops, control statements, etc., in shell scripting to solve problems.
4. Use various utilities of the OS.

5. Manage users and file systems.

Unit	Content	No. of Lectures
1	Introduction: What is linux/unix Operating systems, Difference between linux/unix and other operating systems, Features and Architecture, Various Distributions available in the market, Installation, Booting and shutdown process, System processes (an overview), External and internal commands, Creation of partitions in OS, Processes and its creation phases – Fork, Exec, wait.	5
2	User Management and the File System: Types of Users, Creating users, Granting rights, User management commands, File quota and various file systems available, File System Management and Layout, File permissions, Login process, Managing Disk Quotas, Links (hard links, symbolic links).	5
3	Shell introduction and Shell scripting: What is shell and various type of shell, Various editors present in linux, Different modes of operation in vi editor, What is shell script, Writing and executing the shell script, Shell variable (user defined and system variables), System calls, Using system calls, Pipes and Filters, Decision making in Shell Scripts (If else, switch), Loops in shell, Functions, Utility programs, Pattern matching utility.	5
	<p>Practical:</p> <ol style="list-style-type: none"> 1. Write a shell script to check if the number entered at the command line is prime or not. 2. Write a shell script to display calendars of the specified months. 3. Write a shell script to display calendars of the specified range of months. 4. Write a shell script to accept a login name. If not a valid login name displays the message “Entered login name is invalid!”. 5. Write a shell script to display the date in the mm/dd/yy format. 6. Write a shell script to display on the screen a sorted output of whol command along with the total number of users . 7. Write a shell script to display the multiplication table any number, 8. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file. 9. Write a shell script to find the sum of digits of a given number. 10. Write a shell script to merge the contents of three files, sort the contents and then display them page by page. 11. Write a shell script to find the LCD (least common divisor) of two numbers. 12. Write a shell script to perform the tasks of the basic calculator. 13. Write a shell script to find the power of a given number. 14. Write a shell script to find the binomial coefficient $C(n, x)$. 15. Write a shell script to find the permutation $P(n,x)$. 16. Write a shell script to find the greatest number among the three numbers. 17. Write a shell script to find the factorial of a given number. 18. Write a shell script to check whether the number is Armstrong or not. 19. Write a shell script to check whether the file has all the permissions or not. 	1 Credit

Reading list:

- Sumitabha, Das, Unix Concepts And Applications, Tata McGraw-Hill Education, 2006

- Michael Jang RHCSA/ RHCE Red Hat Linux Certification: Exams (Ex200 & Ex300) (Certification Press), 2011
- Nemeth Synder & Hein, Linux Administration Handbook, Pearson Education, 2nd Edition ,2010
- W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, Unix Network Programming, The sockets Networking API, Vol. 1, 3rd Edition,2014