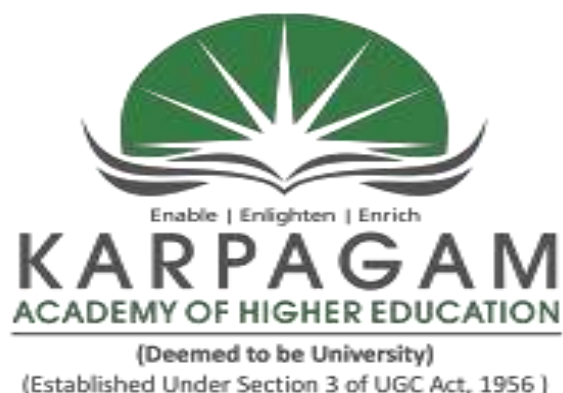


# **B.Sc. COMPUTER SCIENCE**

## **CHOICE BASED CREDIT SYSTEM (CBCS)**

### **Curriculum and Syllabus** **Regular (2017 – 2018)**



**DEPARTMENT OF COMPUTER SCIENCE**

**FACULTY OF ARTS, SCIENCE AND HUMANITIES**

**KARPAGAM ACADEMY OF HIGHER EDUCATION**

**(Deemed to be University)**

**(Established Under Section 3 of UGC Act, 1956)**

**Eachanari (Post), Coimbatore – 641 021.**

**Tamilnadu, India**

**Phone No. 0422-2980011 - 15**

**Fax No: 0422-2980022-23**

**E mail ID: info@karpagam.com**

**Web: www.kahedu.edu.in**

PROGRAM OUTCOMES (POs): The program must enable students to attain by the time of graduation

- a) An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline
- b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
- c) An ability to design, implement and evaluate a computer-based system, process, component or program to meet desired needs.
- d) An ability to function effectively on teams to accomplish a common goal
- e) An understanding of professional, ethical, legal, security and social issues and responsibilities
- f) An ability to communicate effectively with a range of audiences
- g) An ability to use current techniques, skills and tools necessary for computing practice
- h) An ability to use and apply current technical concepts and practices in the core information technologies of human computer interaction, information management, programming, networking and web systems and technologies
- i) An ability to effectively integrate IT-based solutions into the user environment
- j) An understanding of best practices and standards and their application

PROGRAM SPECIFIC OUTCOME (PSOs)

- k) Understand analyze and develop computer programs in the areas related to Database systems and Big data Analytics, cloud computing, soft computing, IoT, Image processing, Green computing, web designing, mobile computing and networking for efficient design of computer based system of varying complexity.
- l) Apply standard software Engineering practices and strategies in software project development using open-source programming environment to deliver a quality for business success.
- m) Be acquainted with the contemporary issues, latest trends in technological development and thereby innovate new ideas and solutions to existing problems.
- n) An ability to produce cost effective, quality and maintainable software products and solutions (services) meeting the global standards and requirements with the knowledge acquired and using the emerging techniques, tools and software engineering methodologies and principles and able to comprehend and write effective project reports in multidisciplinary environment in the context of changing technologies.

## PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO I : To be a working Information Technology (IT) professional with core competencies that can be used on multi-disciplinary projects

PEO II : To understand the importance of relationship building within the IT industry

PEO III : To understand the need for lifelong learning in the exploration and journey in IT

PEO IV : To understand, evaluate and practice ethical behavior within the IT industry

PEO V : To be cognizant of security issues and their impacts on industry

### MAPPING of PEOs and POs

POs	a	b	C	d	e	f	f	h	i	j	k	l	m	n
PEO I	X	X	X				X	X	X				X	
PEO II				X	X	X								X
PEO III	X	X						X		X	X			
PEO IV			X	X	X				X			X		
PEO V					X					X		X		

**KARPAGAM ACADEMY OF HIGHER EDUCATION**  
**(Deemed University Established Under Section 3 of UGC Act, 1956)**

**Coimbatore - 641 021, India**  
**FACULTY OF ARTS, SCIENCE AND HUMANITIES (FASH)**  
**B.Sc Computer Science - Curriculum (CBCS)**  
**(2017 – 2020 Batch)**

Course code	Name of the course	Objectives and outcomes		Instruction hours / week			Credit(s)	Maximum Marks		
		PEOs	POs	L	T	P		CIA	ESE	Total
<b>SEMESTER - I</b>										
17LSU101	Language-I	IV	d,e	04	-	-	4	40	60	100
17CSU101	Programming Fundamentals using C / C++	I	a,b,c	05	-	-	5	40	60	100
17CSU102	Computer System Architecture	I	b,c,g	04	-	-	4	40	60	100
17CSU103	Computer Fundamentals	III	h,j	04	-	-	4	40	60	100
17CSU111	Programming Fundamentals using C / C++ - Practical	I	a,b,c,g	-	-	04	2	40	60	100
17CSU112	Computer System Architecture – Practical	I	a,c,g	-	-	03	2	40	60	100
17CSU113	Computer Fundamentals – Practical	III	b,h,j	-	-	03	2	40	60	100
17AEC101	Environmental Studies	IV	d,e	03	-	-	3	40	60	100
<b>Semester Total</b>				<b>20</b>	<b>-</b>	<b>10</b>	<b>26</b>	<b>320</b>	<b>480</b>	<b>800</b>
<b>SEMESTER – II</b>										
17LSU201	Language – II	IV	d,e	04	-	-	4	40	60	100
17ENU201	English – I	II	d,f	04	-	-	4	40	60	100
17CSU201	Programming in JAVA	I	c,h,i	04	-	-	4	40	60	100
17CSU202	Discrete Structures	III	a,b	04	-	-	4	40	60	100
17CSU203	Multimedia and Applications	IV	e,i	04	-	-	4	40	60	100
17CSU211	Programming in JAVA – Practical	I	a,c,h,i	-	-	04	2	40	60	100
17CSU212	Discrete Structures – Practical	III	a,b,j	-	-	03	2	40	60	100
17CSU213	Multimedia and Applications - Practical	IV	c,e	-	-	03	2	40	60	100
<b>Semester Total</b>				<b>20</b>	<b>-</b>	<b>10</b>	<b>26</b>	<b>320</b>	<b>480</b>	<b>800</b>
<b>SEMESTER - III</b>										
17CSU301	Data Structures	I	a,b,g,h	04	-	-	4	40	60	100
17CSU302	Operating Systems	III	a,b,h,k	04	-	-	4	40	60	100
17CSU303	Computer Networks	III	a,b,j,k	04	-	-	4	40	60	100
17CSU304A	Android Programming	I	a,b,c,m	03	-	-	3	40	60	100

17CSU304B	Programming in MATLAB	IV	c,d,e,i					40	60	100		
17CSU311	Data Structures – Practical	I	a,b,g,h	-	-	04	2	40	60	100		
17CSU312	Operating Systems – Practical	III	a,b,h,k	-	-	04	2	40	60	100		
17CSU313	Computer Networks – Practical	III	a,b,j,k	-	-	04	2	40	60	100		
17CSU314A	Android Programming – Practical	I	a,b,c,m	-	-	03	1	40	60	100		
17CSU314B	Programming in MATLAB - Practical	IV	c,d,e,i	-	-	03		40	60	100		
<b>Semester Total</b>						<b>15</b>	<b>-</b>	<b>15</b>	<b>22</b>	<b>320</b>	<b>480</b>	<b>800</b>
<b>SEMESTER – IV</b>												
17CSU401	Design and Analysis of Algorithms	I	a,b,c,m	04	-	-	4	40	60	100		
17CSU402	Software Engineering	IV	c,d,e,l	04	-	-	4	40	60	100		
17CSU403	Database Management Systems	I	a,b,g,h	04	-	-	4	40	60	100		
17CSU404A	HTML Programming	III	a,b,h,j,k	03	-	-	3	40	60	100		
17CSU404B	XML Programming	III	a,b,h,j,k									
17CSU411	Design and Analysis of Algorithms – Practical	I	a,b,c,m	-	-	04	2	40	60	100		
17CSU412	Software Engineering – Practical	IV	c,d,e,l	-	-	04	2	40	60	100		
17CSU413	Database Management Systems – Practical	I	a,b,g,h	-	-	04	2	40	60	100		
17CSU414A	HTML Programming – Practical	III	a,b,h,j,k			03	1	40	60	100		
17CSU414B	XML Programming – Practical	III	a,b,h,j,k									
<b>Semester Total</b>						<b>15</b>	<b>-</b>	<b>15</b>	<b>22</b>	<b>320</b>	<b>480</b>	<b>800</b>
<b>SEMESTER – V</b>												
17CSU501A	Information Security	I	b,e,m	04	-	-	4	40	60	100		
17CSU501B	Network Programming	I	c,g									
17CSU502A	Microprocessor	III	a,b,h,j	04	-	-	4	40	60	100		
17CSU502B	Digital Image Processing	I	a,b,h									
17CSU503A	Machine Learning	III	A,b,h,k	04	-	-	4	40	60	100		
17CSU503B	Introduction to Data Sciences	II	d,e,f									
17CSU504A	Oracle (SQL/PL-SQL )	IV	c,e,i,l	03	-	-	3	40	60	100		
17CSU504B	Programming in Python	III	b,h,j,k									
17CSU511A	Information Security – Practical	I	b,e,m	-	-	04	2	40	60	100		
17CSU511B	Network Programming – Practical	I	c,g	-	-							
17CSU512A	Microprocessor – Practical	III	a,b,h,j	-	-	04	2	40	60	100		

17CSU512B	Digital Image Processing – Practical	I	a,b,h								
17CSU513A	Machine Learning – Practical	III	a,b,h, k	-	-	04	2	40	60	100	
17CSU513B	Introduction to Data Sciences – Practical	II	d,e,f	-	-						
17CSU514A	Oracle (SQL/PL-SQL ) – Practical	IV	c,e,i,l	-	-	03	1	40	60	100	
17CSU514B	Programming in Python – Practical	III	b,h,j,k								
	<b>Semester Total</b>			<b>15</b>	<b>-</b>	<b>15</b>	<b>22</b>	<b>320</b>	<b>480</b>	<b>800</b>	
<b>SEMESTER –VI</b>											
17CSU601A	Cloud Computing	V	e,j,l	04	-	-	4	40	60	100	
17CSU601B	System Programming	I	a,b,h,i								
17CSU602A	Data Mining	I	a,d,g, m	04	-	-	4	40	60	100	
17CSU602B	Computer Graphics	I	a,c,g, m								
17CSU603A	PHP Programming	III	a,b,h,j, k	03	-	-	3	40	60	100	
17CSU603B	Unix / Linux Programming	IV	c,d,e								
17CSU611A	Cloud Computing – Practical	V	e,j,l	-	-	04	2	40	60	100	
17CSU611B	System Programming – Practical	I	a,b,h,i	-	-	04	2	40	60	100	
17CSU612A	Data Mining – Practical	I	a,d,g, m								
17CSU612B	Computer Graphics – Practical	I	a,c,g, m								
17CSU613A	PHP Programming –Practical	III	a,b,h,j, k	-	-	03	1	40	60	100	
17CSU613B	Unix / Linux Programming – Practical	IV	c,d,e								
17CSU691	Project	II	d,e,f,n	08	-	-	6	40	60	100	
	ECA / NCC / NSS / Sports / General interest etc	Good									
	<b>Semester Total</b>			<b>19</b>	<b>-</b>	<b>11</b>	<b>22</b>	<b>280</b>	<b>420</b>	<b>700</b>	
	<b>Grand Total</b>			<b>104</b>	<b>-</b>	<b>76</b>	<b>140</b>	<b>1880</b>	<b>2820</b>	<b>4700</b>	

Entrepreneur Oriented Courses -Green  
 Employability Oriented Courses -Blue  
 Skill Development Oriented Courses -Red

**கற்பகம் உயர்கல்விகலைக்கழகம்**  
**தமிழ்த்துறை**  
**பகுதி - I தமிழ்ப்பாடத்திட்டம் (2017 - 2018)**  
**இரண்டாம்பருவம்**  
**(இளநிலை அறிவியல் பட்டவகுப்புகளுக்குரியது)**  
**(For I-UG Science Degree Classes) 17LSU201**

**பாடத்திட்டப்பொதுநோக்கம்**

- கற்றல் வழி சிந்தனைத் திறனையும், கருத்து வெளிப்பாட்டுத் திறனையும், மேம்படுத்துதல்.
- ஆய்வுநோக்கை மேம்படுத்துதல்.
- இலக்கியங்கள் உணர்த்தும் வாழ்வின் நுட்பமான பகுதிகளை உணர்த்துதல்.
- மனித மனத்தினைப் பக்குவப்படுத்துதலில் இலக்கியம் தரும் பங்கினை உணர்த்துதல்.
- வளர்ந்து வரும் சமூகத்தில் அறஉணர்வு, பண்பாடு போன்றவை குறித்து அறிவூட்டல்.
- அரசுத் தேர்வுகளுக்கு மாணவர்களை ஆயத்தமாக்குதல்.

**பாடத்திட்டப்பயன்விளைவு**

- இந்தியகுடியரிமைப்பணிமுதலானபோட்டித்தேர்வுகளில், விருப்பப்பாடமாக இடம்பெறுகின்ற, 'தமிழ்இலக்கியவரலாறு' குறித்த முழுமையான அறிமுகம்பெற்றிருத்தல்.
- கல்வெட்டியல், ஓலைச்சுவடியியல் மற்றும் தொல்லியல் சார்ந்த ஆவணத்தேடலுக்குரிய ஆய்வுமனப்பான்மையுடன், இலக்கியங்களை அணுகுதல்.
- தமிழின் வளர்ச்சித்துறையாகிய, 'அறிவியல்தமிழ்' ; 'இணையதமிழ்' குறித்த பன்னோக்கு அணுகுமுறையிலான ஆய்வுச்சிந்தனை மேம்பாடு.
- வேலைவாய்ப்புக்குரிய சுயதிறன் மேம்பாட்டுடன், படைப்பாக்கத்திறன் மேம்பாடும் பெற்றிருத்தல் .
- சமுதாயமற்றும் வாழ்வியல் மதிப்புகளைப் பேணுவதற்குக் கருவியாக இலக்கியங்களை நாடுகின்ற மனப்பான்மை வளர்ச்சி.
- மொழிபெயப்புத்துறை சார்ந்த வேலைவாய்ப்புத்திறன் பெற்றிருத்தல்

**தாள்கள் வரிசையும் தேர்வுச் செயல்திட்டமும் பகுதி-I தமிழ்**

பருவம்	தாள்	கற்பிக்கும் நேரம்/வாரம்	தேர்வு மணிகள்	மதிப்பெண் அக/எழுத்து	மொத்தம்	மதிப்பீடு
இரண்டு	II	4	3	40 / 60	100	4

பகுதி - I, தமிழ்  
17LSU101 :

தமிழ் முதல் தாள்  
(இளநிலை அறிவியல் பட்ட வகுப்புகளுக்குரியது)

பருவம் I

4-H,4-C

அலகு - I : இக்கால இலக்கியம்: (10 மணிநேரம்)

கல்வி : மகாகவி பாரதியார் - சுயசரிதை - ஆங்கிலக் கல்வி.

இன்றைய நிலை : கவிமணி தேசிக விநாயகம் பிள்ளை -  
ஒற்றுமையே உயிர்நிலை.

மனிதநேயம் : கவிஞர் சிற்பி பாலசுப்பிரமணியன் - மலையாளக் காற்று.

சூழலியல் : கவிஞர் வைதீஸ்வரன் - விரல் மீட்டிய மழை.

பெண்ணியம் : கவிஞர் சுகந்தி சுப்பிரமணியம் - புதையுண்ட வாழ்க்கை.

அலகு - II : அற இலக்கியம்: (10 மணிநேரம்)

கொன்றை வேந்தன்: 1-50 பாடல்கள்

திருக்குறள்: பண்புடைமை, வினைத்திட்டம் - 20 குறள்கள்

பழமொழி நானூறு: 5 பாடல்கள்

அலகு - III : சிற்றிலக்கியம்: (10 மணிநேரம்)

மூவருலா: 1-26 கண்ணிகள்

திருச்செந்தூர் முருகன் பிள்ளைத்தமிழ்: 2 பாடல்கள்

கலிங்கத்துப் பரணி: போர்பாடியது - 9 பாடல்கள்

அலகு - IV : கட்டுரை: (10 மணிநேரம்)

1. உயர்தனிச் செம்மொழி - பரிதிமாற்கலைஞர்

2. கட்டிடக்கலை - அ. இராசமாணிக்கனார்

3. வாழ்க்கை - இளவழகனார்

4. ஆளுமைத்திறன் அறிவோம் - ஸ்ரீ கண்ணன்

5. மணற்கேணி - நெ.து.சுந்தரவடிவேலு

அலகு - V : மொழிப்பயிற்சி: (8 மணிநேரம்)

1. பொருத்தமான தமிழ்ச் சொற்களைப் பயன்படுத்துதல்

2. செய்யுள் பொருளுணர் திறன்

3. மொழிபெயர்ப்புப் பயிற்சிகள்

4. கடிதங்கள் மற்றும் விண்ணப்பங்கள் எழுதுதல்

பாட நூல்: கற்பகச்சோலை - தமிழ் ஏடு. கற்பகம் பல்கலைக்கழகத் தமிழ்த் துறை  
வெளியீடு.



**Instruction Hours / week: L: 4 T: 0 P: 0 Marks: Int : 40 Ext : 60 Total: 100**

### Course Objectives

- To impart adequate knowledge on the need of programming languages and problem solving techniques.
- To develop programming skills using the fundamentals and basics of C and C++ Languages
- To enable effective usage of arrays, structures, functions
- To learn effective usage of pointers and to implement the memory management concepts.
- To teach the issues in file organization and the usage of file systems.
- To learn the characteristics of an object-oriented programming language: data abstraction and information hiding, inheritance, and dynamic binding of the messages to the methods.

### Course Outcomes (COs)

After the completion of this course, a successful student will be able to do the following:

1. Develop programs using the basic elements like control statements, Arrays and Strings .
2. understand about the dynamic memory allocation using pointers which is essential for utilizing memory
3. Understand about the code reusability with the help of user defined functions.
4. Learn the basics of file handling mechanism that is essential for understanding the concepts in database management systems.
5. Use the characteristics of an object-oriented programming language in a program.
6. Use the basic object-oriented design principles in computer problem solving.

### UNIT-I

#### 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++. **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). **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)

## UNIT-II

**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.

## UNIT-III

**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. **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

## UNIT-IV

**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. **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 aCSUess in files, Understanding the Preprocessor Directives (#include, #define, #error, #if, #else, #elif, #endif, #ifdef, #ifndef and #undef), Macros.

## UNIT-V

**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. **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) **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, Rethrowing exceptions.

### **Suggested Readings**

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

### **WEB SITES**

1. <http://www.cs.cf.ac.uk/Dave/C/CE.html>
2. <http://www2.its.strath.ac.uk/courses/c/>
3. <http://www.iu.hio.no/~mark/CTutorial/CTutorial.html>
4. <http://www.cplusplus.com/doc/tutorial/>
5. [www.cplusplus.com/](http://www.cplusplus.com/)
6. [www.cppreference.com/](http://www.cppreference.com/)

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

**Course Objectives**

- To enable the students to gain knowledge on the architecture of modern computer.
- To understand how computer stores positive and negative numbers and to perform arithmetic operation of positive and negative numbers.
- To learn about logic gates and solve problems using Boolean algebra.
- To understand the simplification of circuits like adders, subtractors, multiplexers, encoders.
- To understand the basic computer organization and design.
- To learn Cache memory and its importance

**Course Outcomes (COs)**

1. Students will acquire a basic knowledge about computer system architecture, digital circuits and the low - level programming skills.
2. Understand the inner workings and performance capabilities of advanced microprocessors.
3. Solve the problems using Boolean algebra
4. Understand the basic computer organization and design.
5. learn about Cache memory and its importance
6. Solve the binary arithmetic problems and conversion among the number systems

**UNIT -I Introduction**

Logic gates, Boolean algebra, circuit simplification, combinational circuits: Adders and Subtractors – Multiplexers and De multiplexers – Encoders and Decoders- sequential circuits: Flip Flop's, registers, counters and memory units.

**UNIT -II 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

**UNIT –III 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.

**UNIT-IV 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.

## **UNIT –V Memory and Input-Output Organization**

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

### **Suggested Readings:**

1. Dos Reis, A. J. (2009). Assembly Language and Computer Architecture using C++ and JAVA. Course Technology
2. Stallings, W. (2010). Computer Organization and Architecture Designing for Performance (8th ed.) New Delhi: Prentice Hall of India,
3. Mano, M.M. (2013). Digital Design, New Delhi: Pearson Education Asia.
4. Carl Hamacher. (2012). Computer Organization (5th ed.). New Delhi: McGrawHill.

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

**Course Objectives**

- To identify types of computers, how they process information and how individual computers interact with other computing systems and devices.
- To identify the function of computer hardware components.
- To identify different types of software, general concepts relating to software categories, and the tasks to which each type of software is most suited or not suited.
- To identify fundamental concepts relating to database applications.
- To manipulate and control the Windows desktop, files and disks.
- To understand the emerging technologies and their uses.

**Course Outcomes (COs)**

1. Understand the meaning and basic components of a computer system,
2. Gain knowledge about five generations and classification of computer system,
3. Explain the functions of a computer,
4. Identify and discuss the functional units of a computer system,
5. Identify the various input and output units and explain their purposes
6. Understand the emerging technologies and their uses.

**UNIT-I**

**Introduction:** Introduction to computer system, uses, types. **Data Representation:** Number systems and character representation, binary arithmetic. **Human Computer Interface:** Types of software, Operating system as user interface, utility programs.

**UNIT-II**

**Devices:** Input and output devices (with connections and practical demo), keyboard, mouse, joystick, scanner, OCR, OMR, bar code reader, web camera, monitor, printer, plotter.

**UNIT-III**

**Memory:** Primary, secondary, auxiliary memory, RAM, ROM, cache memory, hard disks, optical disks.

**UNIT-IV**

**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.

**UNIT-V**

**Overview of Emerging Technologies:** Bluetooth, cloud computing, big data, data mining, mobile computing and embedded systems.

**Suggested Readings:**

1. Goel, A. (2010). Computer Fundamentals. New Delhi: Pearson Education.
2. Aksoy, P., & DeNardis, L. (2006). Introduction to Information Technology. New Delhi: Cengage Learning
3. Sinha, P. K., & Sinha, P. (2007). Fundamentals of Computers. New Delhi: BPB Publishers.

**Instruction Hours / week: L: 0 T: 0 P: 3    Marks: Int : 40 Ext : 60    Total: 100**

**Course Objectives**

- To impart adequate knowledge on the need of programming languages and problem solving techniques.
- To develop programming skills using the fundamentals and basics of C and C++ Languages
- To enable effective usage of arrays, structures, functions
- To learn effective usage of pointers and to implement the memory management concepts.
- To teach the issues in file organization and the usage of file systems.
- To learn the characteristics of an object-oriented programming language: data abstraction and information hiding, inheritance, and dynamic binding of the messages to the methods.

**Course Outcomes (COs)**

After the completion of this course, a successful student will be able to do the following:

1. Develop programs using the basic elements like control statements, Arrays and Strings .
2. understand about the dynamic memory allocation using pointers which is essential for utilizing memory
3. Understand about the code reusability with the help of user defined functions.
4. Learn the basics of file handling mechanism that is essential for understanding the concepts in database management systems.
5. Use the characteristics of an object-oriented programming language in a program.
6. Use the basic object-oriented design principles in computer problem solving.

**List of Programs**

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 user is Palindrome or not.



6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.

7. WAP to compute the factors of a given number.

8. Write a macro that swaps two numbers. WAP to use it.

9. WAP to print a triangle of stars as follows (take number of lines from user):

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

10. WAP to perform following actions on an array entered by the user:

i) Print the even-valued elements

ii) Print the odd-valued elements

iii) Calculate and print the sum and average of the elements of array

iv) Print the maximum and minimum element of array

v) Remove the duplicates from the array

vi) Print the array in reverse order

The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.

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

12. Write a program that swaps two numbers using pointers.

13. Write a program in which a function is passed address of two variables and then alter its contents.

14. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.

15. Write a program to find sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions or new operator.

16. Write a menu driven program to perform following operations on strings:
- Show address of each character in string
  - Concatenate two strings without using strcat function.
  - Concatenate two strings using strcat function.
  - Compare two strings
  - Calculate length of the string (use pointers)
  - Convert all lowercase characters to uppercase
  - Convert all uppercase characters to lowercase
  - Calculate number of vowels
  - Reverse the string
17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
18. WAP to display Fibonacci series (i)using recursion, (ii) using iteration
19. WAP to calculate Factorial of a number (i)using recursion, (ii) using iteration
20. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.
21. Create Matrix class using templates. Write a menu-driven program to perform following Matrix operations (2-D array implementation):
- Sum
  - Difference
  - Product
  - Transpose
22. Create the Person class. Create some objects of this class (by taking information from the user). Inherit the class Person to create two classes Teacher and Student class. Maintain the respective information in the classes and create, display and delete objects of these two classes (Use Runtime Polymorphism).
23. Create a class Triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator.
24. Create a class Box containing length, breath and height. Include following methods in it:
- Calculate surface Area
  - Calculate Volume
  - Increment, Overload ++ operator (both prefix & postfix)

- d) Decrement, Overload -- operator (both prefix & postfix)
- e) Overload operator == (to check equality of two boxes), as a friend function
- f) Overload Assignment operator
- g) Check if it is a Cube or cuboid

Write a program which takes input from the user for length, breath and height to test the above class.

25. Create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.

26. Write a program to retrieve the student information from file created in previous question and print it in following format:

Roll No. Name Marks

27. Copy the contents of one text file to another file, after removing all whitespaces.

28. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.

29. Write a program that will read 10 integers from user and store them in an array. Implement array using pointers. The program will print the array elements in ascending and descending order.

---

Instruction Hours / week: L: 0 T: 0 P: 3    Marks: Int : 40 Ext : 60    Total: 100

### Course Objectives

- To enable the students to gain knowledge on the architecture of modern computer.
- To understand how computer stores positive and negative numbers and to perform arithmetic operation of positive and negative numbers.
- To learn about logic gates and solve problems using Boolean algebra.
- To understand the simplification of circuits like adders, subtractors, multiplexers, encoders.
- To understand the basic computer organization and design.
- To learn Cache memory and its importance

### Course Outcomes (COs)

1. Students will acquire a basic knowledge about computer system architecture, digital circuits and the low - level programming skills.
2. Understand the inner workings and performance capabilities of advanced microprocessors.
3. Solve the problems using Boolean algebra
4. Understand the basic computer organization and design.
5. learn about Cache memory and its importance
6. Solve the binary arithmetic problems and conversion among the number systems

### List of Experiments

(Any 8 Experiments)

1. Verification of Logic Gates
2. Code converters
3. Realization of Multiplexer using basic gates
4. Encoder and Decoder
5. Realization Half and Full adders
6. Realization of Subtractor
7. Realization of Parity generator
8. Flip-Flop Circuits
9. Digital to analog Converters
10. Demonstrate a Basic Arithmetic Computing operations

---

**Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60**

**Total: 100**

### Course Objectives

- Create a document in Microsoft Word with formatting options.
- Create, edit, save, and print documents to include documents with lists and tables.
- Add a header and footer to a document and add a graphic to a document.
- Write functions in Microsoft Excel to perform basic calculations and to convert number to text and text to number.
- Construct formulas, including the use of built-in functions, and relative and absolute references.
- Create and modify charts.

### Course Outcomes (COs)

1. Modify text using various formatting options from the editing tools under the Home tab
2. Set up section breaks to create different headers and footers for the odd and even pages within the document sections.
3. Demonstrate the mechanics and uses of Word tables to organize and present data.
4. Demonstrate working knowledge of using Word's themes and clip art to create a variety of visual effects.
5. Create and design a spreadsheet for general office use.
6. Demonstrate the use of basic functions and formulas in Excel

Practical exercises based on MS Office/ Open Office tools using document preparation and spreadsheet handling packages.

#### MS Word

1. Prepare a **grocery list** having four columns (Serial number, The name of the product, quantity and price) for the month of April, 06.

- 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.
- 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 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.

- a. Compose a note inviting friends to a get-together at your house, including a list of things to bring with them.
- b. Design a certificate in landscape orientation with a border around the document.
- c. Design a Garage Sale sign.
- d. 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.

<b>Salesperson</b>	<b>Dolls</b>	<b>Trucks</b>	<b>Puzzles</b>
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

<b>REGIONAL SALES PROJECTION</b>						
<b>State</b>	<b>Qtr1</b>	<b>Qtr2</b>	<b>Qtr3</b>	<b>QTR4</b>	<b>Qtr Total</b>	<b>Rate Amount</b>
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

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
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
$\geq 80$	A+
$\geq 60 < 80$	A
$\geq 50 < 60$	B
$< 50$	F



3. Given the following worksheet

	A	B	C	D	E	F	
1	<b>Salesman</b>		<b>Sales in (Rs.)</b>				
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 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:

#### Allowances

- HRA Dependent on Basic
  - 30% of Basic if Basic ≤ 1000
  - 25% of Basic if Basic > 1000 & Basic ≤ 3000
  - 20% of Basic if Basic > 3000
- DA Fixed for all employees, 30% of Basic
- Conveyance Allowance Rs. 50/- if Basic is ≤ 1000  
Rs. 75/- if Basic > 1000 & Basic ≤ 2000  
Rs. 100 if Basic > 2000
- Entertainment Allowance NIL if Basic is ≤ 1000  
Rs. 100/- if Basic > 1000

#### Deductions

- Provident Fund 6% of Basic
- Group Insurance Premium Rs. 40/- if Basic is ≤ 1500  
Rs. 60/- if Basic > 1500 & Basic ≤ 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 Installments	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

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

7. The following table gives 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

- Calculate total sale year wise.
- Calculate the net sale made by each salesman
- Calculate the maximum sale made by the salesman
- Calculate the commission for each salesman under the condition.
  - If total sales >4,00,000 give 5% commission on total sale made by the salesman.
  - Otherwise give 2% commission.
- Draw a bar graph representing the sale made by each salesman.
- Draw a pie graph representing the sale made by salesman in 2000.

8. Enter the following data in Excel Sheet

### PERSONAL BUDGET FOR FIRST QUARTER

Monthly Income (Net): 1,475

EXPENSES	JAN	FEB	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.

- Calculate Monthly total.
- Surplus = Monthly income - Monthly total.
- What would be total surplus if monthly income is 1500.
- How much does telephone expense for March differ from quarter average.
- 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	

- Compute the total revenue earned.
- Plot the line chart to compare the revenue of all publisher for 4 years.
- Chart Title should be Total Revenue of sam's Bookstall (1997-2000)
- Give appropriate categories and value axis title.

10. Generate 25 random numbers between 0 & 100 and find their sum, average and count. How many no. are in range 50-60.

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Internal: 100****Total: 100****Course Objectives**

- To create the awareness about environmental problems among people.
- To develop an attitude of concern for the environment.
- To motivate public to participate in environment protection and improvement.
- To understand the transnational character of environmental problems and ways of addressing them, including interactions across local to global scales.
- To apply systems concepts and methodologies to analyze and understand interactions between social and environmental processes.
- To reflect critically about their roles and identities as citizens, consumers and environmental actors in a complex, interconnected world.

**Course Outcomes (COs)**

1. Master core concepts and methods from ecological and physical sciences and their application in environmental problem solving.
2. Master core concepts and methods from economic, political, and social analysis as they pertain to the design and evaluation of environmental policies and institutions.
3. Appreciate the ethical, cross-cultural, and historical context of environmental issues and the links between human and natural systems.
4. Understand the transnational character of environmental problems and ways of addressing them, including interactions across local to global scales.
5. Apply systems concepts and methodologies to analyze and understand interactions between social and environmental processes.
6. Reflect critically about their roles and identities as citizens, consumers and environmental actors in a complex, interconnected world.

**UNIT-I:** Environment Definition, scope and importance, components, Ecosystem Definition, Concept, Scope, importance, Structure and functions of ecosystem. Energy flow, Ecological succession Food chains and food webs. Classification of ecosystem.

**Unit II: Natural Resources - Renewable and Non-renewable Resources:** Natural resources and associated problems. Forest resources, Water resources, Mineral resources, Food resources, Energy resources, Land resources : Use and over-utilization, exploitation. Role of an individual in conservation of natural resources. Equitable use of resources for sustainable lifestyles. Ill-effects of fire works.

**Unit III: Biodiversity and Its Conservation:** Introduction, definition: genetic, species and ecosystem diversity. Biogeographical classification of India. Value of biodiversity: consumptive use, productive use, social, ethical, aesthetic and option values. Biodiversity at global, National and local levels. India as a mega-diversity nation. Hot-spots of biodiversity. Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife

conflicts. Endangered and endemic species of India. Conservation of biodiversity: in-situ and ex-situ conservation of biodiversity.

**Unit IV: Environmental Pollution** - Definition, Causes, effects and control measures of Air pollution, Water pollution, Soil pollution, Marine pollution, Noise pollution, Thermal pollution, Nuclear hazards, Solid waste management: Causes, effects and control measures of urban and industrial wastes. Role of an individual in prevention of pollution. Pollution case studies. Disaster management: Floods, earthquake, cyclone and landslides.

**Unit V: Social Issues and the Environment:** From unsustainable to sustainable development. Urban problems related to energy. Water conservation, rain water harvesting, watershed management. Resettlement and rehabilitation of people; its problems and concerns. Environmental ethics: Issues and possible solutions. Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust. Case studies. Wasteland reclamation. Consumerism and waste products. Environment Protection Act. Air (Prevention and Control of Pollution) Act. Water (Prevention and Control of Pollution) Act. Wildlife Protection Act. Forest Conservation Act. Issues involved in enforcement of environmental legislation. Public awareness. Population growth, variation among nations. Population explosion—Family Welfare Programme. Environment and human health. Human rights. Value education. HIV/AIDS. Women and Child Welfare. Role of Information Technology in environment and human health.

### **Suggested Readings**

T1: Tripathy, S.N., & Sunakar Panda. (2004). Fundamentals of Environmental Studies (2<sup>nd</sup> ed.) . New Delhi: Vrianda Publications Private Ltd.

T2: Arvind Kumar. (2004). A Textbook of Environmental Science. New Delhi: APH Publishing Corporation.

T3: Verma, P.S., & Agarwal V.K. (2001). Environmental Biology (Principles of Ecology) . New Delhi: S.Chand and Company Ltd.

T4: Anubha Kaushik., & Kaushik, C.P. (2004). Perspectives in Environmental Studies. New Delhi: New Age International Pvt. Ltd. Publications.

R1: Singh, M.P., Singh, B.S., & Soma, S. Dey. (2004). Conservation of Biodiversity and Natural Resources. New Delhi: Daya Publishing House.

R2: Daniel, B. Botkin., & Edward, A. Keller. (1995). Environmental Science New York: John Wiley and Sons, Inc.,.

R3: Uberoi, N.K. (2005). Environmental Studies. New Delhi: Excel Books Publications.

**கற்பகம் உயர்கல்விகலைக்கழகம்**  
**தமிழ்த்துறை**  
**பகுதி - I தமிழ்ப்பாடத்திட்டம் (2017 - 2018)**  
**இரண்டாம்பருவம்**  
**(இளநிலை அறிவியல் பட்டவகுப்புகளுக்குரியது)**  
**(For I-UG Science Degree Classes) 17LSU201**

**பாடத்திட்டப்பொதுநோக்கம்**

- கற்றல் வழி சிந்தனைத் திறனையும், கருத்து வெளிப்பாட்டுத் திறனையும், மேம்படுத்துதல்.
- ஆய்வுநோக்கைமேம்படுத்துதல்.
- இலக்கியங்கள் உணர்த்தும் வாழ்வின் நுட்பமான பகுதிகளை உணர்த்துதல்.
- மனித மனத்தினைப் பக்குவப்படுத்துதலில் இலக்கியம் தரும் பங்கினை உணர்த்துதல்.
- வளர்ந்து வரும் சமூகத்தில் அறஉணர்வு, பண்பாடு போன்றவை குறித்து அறிவூட்டல்.
- அரசுத் தேர்வுகளுக்கு மாணவர்களை ஆயத்தமாக்குதல்.

**பாடத்திட்டப்பயன்விளைவு**

- இந்தியகுடியரிமைப்பணிமுதலானபோட்டித்தேர்வுகளில், விருப்பப்பாடமாக இடம்பெறுகின்ற, 'தமிழ்இலக்கியவரலாறு' குறித்தமுழுமையான அறிமுகம்பெற்றிருத்தல்.
- கல்வெட்டியல், ஓலைச்சுவடியியல் மற்றும் தொல்லியல் சார்ந்த ஆவணத்தேடலுக்குரிய ஆய்வுமனப்பான்மையுடன், இலக்கியங்களை அணுகுதல்.
- தமிழின் வளர்ச்சித்துறையாகிய, 'அறிவியல்தமிழ்' ; 'இணையதமிழ்' குறித்த பன்னோக்கு அணுகுமுறையிலான ஆய்வுச்சிந்தனை மேம்பாடு.
- வேலைவாய்ப்புக்குரிய சுயதிறன் மேம்பாட்டுடன், படைப்பாக்கத்திறன் மேம்பாடும் பெற்றிருத்தல் .
- சமுதாயமற்றும் வாழ்வியல் மதிப்புகளைப் பேணுவதற்குக் கருவியாக இலக்கியங்களை நாடுகின்ற மனப்பான்மை வளர்ச்சி.
- மொழிபெயப்புத்துறை சார்ந்த வேலைவாய்ப்புத்திறன் பெற்றிருத்தல்

**தாள்கள் வரிசையும் தேர்வுச்செயல்திட்டமும் பகுதி-I தமிழ்**

பருவம்	தாள்	கற்பிக்கும் நேரம்/வாரம்	தேர்வு மணிகள்	மதிப்பெண் அக/எழுத்து	மொத்தம்	மதிப்பீடு
இரண்டு	II	4	3	40 / 60	100	4

பகுதி - I, தமிழ்

பருவம் II

17LSU201 :

தமிழ் இரண்டாம் தாள் 4-H,4-C  
(இளநிலை அறிவியல் பட்ட வகுப்புகளுக்குரியது)

அலகு - I : பக்தி இலக்கியம் (10 மணிநேரம்)

சைவ, வைணவ இலக்கியங்கள் - தோற்றம் ,வளர்ச்சி, வரலாறு.

1. சைவம் - பெரியபுராணம் - திருமூலநாயனார் புராணம்.

2. வைணவம் - பெரியாழ்வார் திருமொழி: 10 பாடல்கள்.

அலகு - II : சங்க இலக்கியம் : (15 மணிநேரம்)

சங்க இலக்கியங்கள் அறிமுகம்

அ). எட்டுத்தொகை

நற்றிணை : பிரசம் கலந்த - பாலை -110

குறுந்தொகை : கருங்கட்டாக் கலை - குறிஞ்சி- 69

ஐங்குறுநூறு : நெய்தல்-தொண்டிப்பத்து:

திரைஇமிழ் இன்னிசை-171

பதிற்றுப்பத்து : சிதைந்தது மன்ற - 27

பரிபாடல்: பரிபாடல் திரட்டு-மதுரை நகர்ச்சிறப்பு -

உலகம் ஒரு நிறையாத்தான்-6, மாயோன் கொப்பூழ்-7, செய்யாட்கு

இழைத்த-9, கார்த்திகை காதில்-10, ஈவாரைக் கொண்டாடி-11.

கலித்தொகை : சுடர்தொட கேளாய்: குறிஞ்சிக்கலி- 36

அகநானூறு : அன்னாய் வாழி வேண்டன்னை - குறிஞ்சி - 48

புறநானூறு : யாதும் ஊரே யாவருங் கேளிர் -பொதுவியல்- 192

ஆ). பத்துப்பாட்டு

திருமுருகாற்றுப்படை - பழமுதிர்ச்சோலையின் சிறப்பு

முருகன் இருப்பிடங்கள் - 'சிறுதினை மலரொடு' என்பதிலிருந்துதொடங்கி,

'அறிந்தவாறே' என்பது வரையிலான தொடர்கள்: 218-249.

முருகன் அருள்புரிதல் - 'தெய்வம் சான்ற' என்பதிலிருந்து தொடங்கி,

'நல்குமதி' என்பது வரையிலான தொடர்கள்: 286-295.

அலகு - III : காப்பியம்

(6 மணிநேரம்)

**சிலப்பதிகாரம்:**

**மங்கல வாழ்த்துப் பாடல்: (21-29) – கண்ணகியின் சிறப்பு:**

நாகநீள் நகரொடு' என்பதிலிருந்து தொடங்கி,

'கண்ணகி என்பாண் மன்னோ' என்பது வரையிலான தொடர்கள்.

**நடுகற்காதை: (207-234) - சேரன் செங்குட்டுவன் கண்ணகிக்குக் கோயில்**

**எடுத்தல்:** 'அருந்திறலரசர்' என்பதிலிருந்து தொடங்கி, 'மன்னவரேறென்' என்பது வரையிலான தொடர்கள்.

**வாழ்த்துக்காதை: (482-485) - செங்குட்டுவனுக்குக் கண்ணகி**

**காட்சியளித்தல்:** 'என்னே' என்பதிலிருந்து தொடங்கி, 'விசும்பில் தோன்றுமால்' என்பது வரையிலான தொடர்கள்.

**வழக்குரை காதை: பத்தினிப் பெண்டிர் எழுவர் கதை:** 'நீர்வார் கண்ணை' என்பதிலிருந்து தொடங்கி, 'புகாரென் பதியே' என்பது வரையிலான தொடர்கள்.

**வஞ்சினமாலை:** 'வன்னி மரமும்' என்பதிலிருந்து தொடங்கி, 'பதிப்பிறந்தேன்' என்பது வரையிலான தொடர்கள்.

**அலகு - IV : சிறுகதை**

**(10 மணிநேரம்)**

1. குளத்தங்கரை அரசமரம் – வ.வே.சு.ஐயர்
2. காட்டில் ஒரு மான் - அம்பை
3. நாற்காலி – கி.ராஜநாராயணன்
4. நகரம் – சுஜாதா

**அலகு- V : மொழிப்பயிற்சி**

**(7 மணிநேரம்)**

படைப்பிலக்கியப் பயிற்சிகள் (கதை, கவிதை, கட்டுரை, உரைநடை)

மொழிபெயர்ப்பு [[

**பாட நூல்:** கற்பகச்சோலை – தமிழ் ஏடு. கற்பகம் பல்கலைக்கழகத் தமிழ்த் துறை வெளியீடு.



---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

(For all undergraduate students admitted from 2016 onwards)

**Course Objectives:**

- To help students enhance their Language skills
- To introduce different kinds of literary works
- To familiarize different genres of Literature
- To instruct moral values through literature.
- To improve their productive and receptive skills
- To strengthen the basic knowledge about grammar

**Course Outcomes:**

1. Develop the four types of skills
2. Reading and comprehending literary works
3. Genres of literature to provide moral education
4. Develop communication skills in business environment
5. Interpersonal skills will be developed.
6. Betterment of language competence

**UNIT - I : PROSE**

1. Morals in the Indian Context - Francis Nicholas Chelliah
2. How Comic Books help us to relive our Childhood - Benoit Peeters
3. Let's Do What India Needs From Us - Dr.A.P.J. Abdul Kalam

**UNIT - II : POEM**

1. The Stolen Boat - William Wordsworth
2. A River - A.K. Ramanujan
3. The Sailor - Safaa Fathy

**UNIT - III : SHORT STORIES**

1. Rapunzel - Brothers Grimm
2. The Romance of a Busy Broker - O.Henry
3. The Nightingale and the Rose - Oscar Wilde.

**UNIT - IV**

- GRAMMAR :**
1. Tenses
  2. Auxiliaries (Primary and Modal)
  3. Articles
  4. Tag Questions

## **UNIT - V**

### **FUNCTIONAL ENGLISH**

1. Filling the blanks with the suitable form of verb in a conditional sentence.
2. Dialogue Writing
3. Changing positive to negative without altering the meaning
4. Fill in the blank with suitable modal
5. Framing a question to a statement
6. Rewrite the sentences changing the underlined word as directed

**Prescribed Text:** Reminisce, Published by the Department of English, Karpagam Academy of Higher Education.

**Suggested Reading:** Hewings Martin, 1999 Advanced English Grammar, Cambridge University Press.

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

### Course Objectives

- To understand the fundamentals of programming such as variables, conditional and iterative execution, methods, etc.
- To understand fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.
- To use the Java SDK environment to create, debug and run simple Java programs.
- To use Java in various technologies in different platforms.
- To understand the fundamental of Packages and access modifiers and interface in java.
- To understand the fundamental of Exception Handling and AWT component and AWT classes.

### Course Outcomes (COs)

1. Student will obtain knowledge of the structure and model of the Java programming language.
2. How to use the Java programming language for various programming technologies (understanding)
3. Develop software in the Java programming language (application)
4. Evaluate user requirements for software functionality required to decide whether the Java programming language can meet user requirements (analysis)
5. Propose the use of certain technologies by implementing them in the Java programming language to solve the given problem (synthesis)
6. Choose an engineering approach to solving problems, starting from the acquired knowledge of programming and knowledge of operating systems. (evaluation)

### UNIT-I

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

### UNIT-II

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

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

### **UNIT-III**

#### **Inheritance, Interfaces, Packages, Enumerations, Autoboxing and Metadata**

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

### **UNIT-IV**

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

### **UNIT-V**

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

#### **Suggested Readings:**

1. Ken Arnold., James Gosling., & David Homes. (2005). The Java Programming Language (4th ed.).
2. James Gosling., Bill Joy., Guy, L. Steele Jr., Gilad Bracha., & Alex Buckley. (2014).The Java Language Specification, Java SE (8 ed.). Addison Wesley.
3. Joshua Bloch. (2008). Effective Java (2nd ed.). Addison-Wesley.
4. Cay, S. Horstmann., GaryCornell. (2012).Core Java 2 Volume 1 (9th ed.). . New Delhi: Prentice Hall.
5. Cay, S. Horstmann., Gary Cornell. (2013). Core Java 2 Volume 2 - Advanced Features(9th ed.). New Delhi: Printice Hall.
6. Bruce Eckel. (2002). Thinking in Java (3rd ed.). New Delhi: PHI.
7. Balaguruswamy, E. (2009). Programming with Java (4th ed.). New Delhi: McGraw Hill.
8. Paul Deitel., & Harvey Deitel. (2011). Java: How to Program (10th ed.). New Delhi: Prentice Hall.
9. (2005).Head First Java (2nd ed.). Orielly Media Inc.
10. David, J. Eck. (2009). Introduction to Programming Using Java. New Delhi: CreateSpace Independent Publishing Platform.

11. John , R. Hubbard. (2004). Programming with JAVA, Schaum's Series, (2nd ed.).

**WEB SITES**

1. [java.sun.com/docs/books/tutorial/](http://java.sun.com/docs/books/tutorial/)
2. [www.en.wikipedia.org/wiki/Java](http://www.en.wikipedia.org/wiki/Java)
3. [www.java.net/](http://www.java.net/)

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

### Course Objectives

This course enables the students to

- To learn the basic concepts of sets, types of sets, functions and relations
- To understand about Pigeonhole principle, Permutation and Combination, Mathematical Induction
- To solve the problems using Recurrence relations and generating functions.
- To know the basic concepts of Logical Connectives, Graphs and Trees.
- To express ideas using mathematical notation
- To solve problems with the help of tools of mathematical analysis.

### Course Outcomes

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

1. Familiar with elementary algebraic set theory.
2. Acquire a fundamental understanding of the core concepts in growth of functions.
3. Describe the method of recurrence relations.
4. Get wide knowledge about graphs and trees
5. Initiate to knowledge from inference theory
6. Solve problems with the help of tools of mathematical analysis

### UNIT I

**Sets:** Introduction, Sets , finite and infinite sets, uncountably 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.

### UNIT II

**Growth of Functions:** Asymptotic Notations, Summation formulas and properties, Bounding Summations, approximation by Integrals

### UNIT III

**Recurrences:** Recurrence relations, generating functions, linear recurrence relations with constant coefficients and their solution, Substitution Method, recurrence trees, Master theorem.

### UNIT IV

**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

### UNIT V

**Propositional Logic:** Logical Connectives, Well-formed Formulas, Tautologies, Equivalences, Inference Theory.

## SUGGESTED READINGS

1. Kenneth Rosen. (2006). Discrete Mathematics and Its Applications (6th ed.). New Delhi: McGraw Hill.
2. Tremblay , J .P. , & Manohar, R. (1997). Discrete Mathematical Structures with Applications to Computer Science. New Delhi: McGraw-Hill Book Company.
3. Coremen, T.H., Leiserson, C.E. , & R. L. Rivest. (2009). Introduction to algorithms, (3rd ed.). New Delhi: Prentice Hall on India.
4. 3. Albertson, M. O.,& Hutchinson, J. P. (1988). Discrete Mathematics with Algorithms . New Delhi: John wiley Publication.
5. Hein, J. L. (2009). Discrete Structures, Logic, and Computability( 3rd ed.). New Delhi: Jones and Bartlett Publishers.
6. Hunter, D.J. (2008). Essentials of Discrete Mathematics. New Delhi: Jones and Bartlett Publishers.

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

### Course Objectives

- To learn and understand technical aspect of Multimedia Systems.
- To understand the standards available for different audio, video and text applications.
- To Design and develop various Multimedia Systems applicable in real time.
- To learn various multimedia authoring systems.
- To understand various networking aspects used for multimedia applications.
- To develop multimedia application and analyze the performance of the same.

### Course Outcomes (COs)

1. Developed understanding of technical aspect of Multimedia Systems.
2. Understand various file formats for audio, video and text media.
3. Develop various Multimedia Systems applicable in real time.
4. Design interactive multimedia software.
5. Apply various networking protocols for multimedia applications.
6. To evaluate multimedia application for its optimum performance.

### UNIT-I

**Multimedia:** Introduction to multimedia, components, uses of multimedia, multimedia applications, virtual reality. **Text:** Fonts & Faces, Using Text in Multimedia, Font Editing & Design Tools, Hypermedia & Hypertext.

### UNIT-II

**Images:** Still Images – bitmaps, vector drawing, 3D drawing & rendering, natural light & colors, computerized colors, color palettes, image file formats. **Sound:** Digital Audio, MIDI Audio, MIDI vs Digital Audio, Audio File Formats.

### UNIT-III

**Video:** How video works, analog video, digital video, video file formats, video shooting and editing. **Animation:** Principle of animations, animation techniques,

### UNIT-IV

**Animation:** animation file formats. **Internet and Multimedia:** www and HTML, multimedia on the web – web servers, web browsers, web page makers and site builders.

### UNIT-V

**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.



**Suggested Readings:**

1. Tay Vaughan. (2011). Multimedia: Making it work (8th ed.). New Delhi: TMH,
2. Ralf Steinmetz., & Klara Naharstedt. (2012).Multimedia: Computing, Communications Applications. New Delhi: Pearson.
3. Keyes. (2000). Multimedia Handbook. New Delhi: TMH.
4. Andleigh, K., & Thakkar, K. (2013).Multimedia System Design. New Delhi: PHI.

**WEB SITES**

1. [en.wikipedia.org/wiki/Multimedia](http://en.wikipedia.org/wiki/Multimedia)
2. [www.arena-multimedia.com/](http://www.arena-multimedia.com/) -
3. [www.nextwavemultimedia.com/](http://www.nextwavemultimedia.com/)

---

**Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Int : 40 Ext : 60 Total: 100**

**Course Objectives**

- To understand the fundamentals of programming such as variables, conditional and iterative execution, methods, etc.
- To understand fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.
- To use the Java SDK environment to create, debug and run simple Java programs.
- To use Java in various technologies in different platforms.
- To understand the fundamental of Packages and access modifiers and interface in java.
- To understand the fundamental of Exception Handling and AWT component and AWT classes.

**Course Outcomes (COs)**

1. Student will obtain knowledge of the structure and model of the Java programming language.
2. How to use the Java programming language for various programming technologies (understanding)
3. Develop software in the Java programming language (application)
4. Evaluate user requirements for software functionality required to decide whether the Java programming language can meet user requirements (analysis)
5. Propose the use of certain technologies by implementing them in the Java programming language to solve the given problem (synthesis)
6. Choose an engineering approach to solving problems, starting from the acquired knowledge of programming and knowledge of operating systems. (evaluation)

**List of Programs**

1. To find the sum of any number of integers entered as command line arguments
2. To find the factorial of a given number
3. To learn use of single dimensional array by defining the array dynamically.
4. To learn use of .length in case of a two dimensional array
5. To convert a decimal to binary number
6. To check if a number is prime or not, by taking the number as input from the keyboard
7. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument
8. Write a program that show working of different functions of String and StringBuffer class like setCharAt(), setLength(), append(), insert(), concat() and equals().

9. Write a program to create a `—distance` class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer
10. Modify the `—distance` class by creating constructor for assigning values (feet and inches) to the distance object. Create another object and assign second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.
11. Write a program to show that during function overloading, if no matching argument is found, then java will apply automatic type conversions(from lower to higher data type)
12. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword
13. Write a program to show the use of static functions and to pass variable length arguments in a function.
14. Write a program to demonstrate the concept of boxing and unboxing.
15. Create a multi-file program where in one file a string message is taken as input from the user and the function to display the message on the screen is given in another file (make use of Scanner package in this program).
16. Write a program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate fibonacci series is given in a different file belonging to the same package.
17. Write a program that creates illustrates different levels of protection in classes/subclasses belonging to same package or different packages
18. Write a program `—DivideByZero` that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.
19. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.
20. Write a program to create your own exception types to handle situation specific to your application (Hint: Define a subclass of Exception which itself is a subclass of Throwable).
21. Write a program to demonstrate priorities among multiple threads.
22. Write a program to demonstrate multithread communication by implementing synchronization among threads (Hint: you can implement a simple producer and consumer problem).
23. Write a program to create URL object, create a URLConnection using the `openConnection()` method and then use it examine the different components of the URL and content.
24. Write a program to implement a simple datagram client and server in which a message that is typed into the server window is sent to the client side where it is displayed.
25. Write a program that creates a Banner and then creates a thread to scrolls the message in the banner from left to right across the applet's window.
26. Write a program to get the URL/location of code (i.e. java code) and document(i.e. html file).

27. Write a program to demonstrate different mouse handling events like `mouseClicked()`, `mouseEntered()`, `mouseExited()`, `mousePressed`, `mouseReleased()` and `mouseDragged()`.
28. Write a program to demonstrate different keyboard handling events.
29. Write a program to generate a window without an applet window using `main()` function.
30. Write a program to demonstrate the use of push buttons.

---

**Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Int : 40 Ext : 60**
**Total: 100****Course Objectives**

This course enables the students to

- To learn the basic concepts of sets, types of sets, functions and relations
- To understand about Pigeonhole principle, Permutation and Combination, Mathematical Induction
- To solve the problems using Recurrence relations and generating functions.
- To know the basic concepts of Logical Connectives, Graphs and Trees.
- To express ideas using mathematical notation
- To solve problems with the help of tools of mathematical analysis.

**Course Outcomes**

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

1. Familiar with elementary algebraic set theory.
2. Acquire a fundamental understanding of the core concepts in growth of functions.
3. Describe the method of recurrence relations.
4. Get wide knowledge about graphs and trees
5. Initiate to knowledge from inference theory
6. Solve problems with the help of tools of mathematical analysis

**List of programs**

1. Write a C Program to find the number of subsets of a set contains n elements.
2. Write a C Program to find transitive closure of a relation.
3. Write a C Program to prove  
 $1/(1*2) + 1/(2*3) + \dots + 1/(n(n+1)) = n/(n+1)$
4. Write a C Program to to perform the sum =  $1 + (1+2) + (1+2+3) + \dots + (1+2+\dots+n)$
5. Write a C program to print Fibonacci series till Nth term using recursion
6. Write a C program in c to calculate factorial of a number using recursion
7. Write a C Program to find a minimum spanning tree using Prim's algorithm
8. Write a C program to find the shortest path with the lower cost in a graph using Dijkstra's Algorithm
9. Write a C Program to construct the truth table for the following formula.  
 (i)  $P \wedge Q \wedge R$     (ii)  $P \wedge Q \wedge R$     (iii)  $P \wedge Q \wedge R$
10. Write a C Program to prove De – Morgan's law.

**Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60 Total: 100**

### **Course Objectives**

- To learn and understand technical aspect of Multimedia Systems.
- To understand the standards available for different audio, video and text applications.
- To Design and develop various Multimedia Systems applicable in real time.
- To learn various multimedia authoring systems.
- To understand various networking aspects used for multimedia applications.
- To develop multimedia application and analyze the performance of the same.

### **Course Outcomes (COs)**

1. Developed understanding of technical aspect of Multimedia Systems.
2. Understand various file formats for audio, video and text media.
3. Develop various Multimedia Systems applicable in real time.
4. Design interactive multimedia software.
5. To evaluate multimedia application for its optimum performance.
6. Design different application in M.M and use different tools like Adobe Photoshop and flash.

Practical exercises based on concepts listed in theory using Flash/ GIMP/ PhotoShop/ Animation Tools/ Image Editors/ Video Editors.

### **Implement the followings using Flash-**

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

7. Create an animation in which text Hello gets converted into GoodBye (using motion/shape tweening).
8. Create an animation having five images having fade-in fade-out effect.
9. Create an scene to show the sunrise (using multiple layers and motion tweening) .
10. Create an animation to show the ripple effect.
11. Create an animation (using Shape tweening and shape hints) for transforming one shape into another.
12. Create an animation for bouncing ball (you may use motion guide layer).

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

**Course Objectives**

- To understand the fundamental concepts of data structures
- To Learn linear data structures – lists, stacks, and queues
- To apply Tree and Graph structures
- To understand and apply sorting, searching algorithms
- To know about hashing algorithms
- To develop application using data structures

**Course Outcomes (COs)**

Upon completion of this course, the student will be able to:

1. Implement abstract data types for linear data structures.
2. Apply the different linear and non-linear data structures to problem solutions.
3. Analyze the applications of tree.
4. Implement graph theory over various data structures.
5. Critically analyze the various sorting algorithms.
6. Apply searching algorithms over various data structures.

**UNIT-I**

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

**UNIT-II**

Linked Lists Singly, Doubly and Circular Lists (Array and Linked representation); Normal and Circular, representation of Stack in Lists; Self Organizing Lists; Skip Lists Queues, Array and Linked representation of Queue, De-queue, Priority Queues

**UNIT-III**

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

**UNIT-IV**

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



## UNIT-V

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

### Suggested Readings

1. Adam Drozdek. (2012). Data Structures and algorithm in C++(3rd ed.). New Delhi: Cengage Learning.
2. Sartaj Sahni. (2011). Data Structures, Algorithms and applications in C++(2nd ed.). New Delhi: Universities Press.
3. Aaron, M. Tenenbaum., Moshe, J. Augenstein., & Yedidiah Langsam. (2009). Data Structures Using C and C++(2nd ed.). New Delhi: PHI.
4. Robert, L. Kruse. (1999). Data Structures and Program Design in C++. New Delhi: Pearson.
5. Malik, D.S. (2010). Data Structure using C++(2nd ed.). New Delhi: Cengage Learning,.
6. Mark Allen Weiss. (2011). Data Structures and Algorithms Analysis in Java (3rd ed.). New Delhi: Pearson Education.
7. Aaron, M. Tenenbaum., Moshe, J. Augenstein., & Yedidiah Langsam. (2003). Data Structures Using Java. New Delhi: PHI.
8. Robert Lafore. (2003). Data Structures and Algorithms in Java(2<sup>nd</sup> ed.). New Delhi: Pearson/ Macmillan Computer Pub.
9. John Hubbard. (2009). Data Structures with JAVA(2nd ed.) . New Delhi: McGraw Hill Education (India) Private Limited.
10. Goodrich, M., & Tamassia, R. (2013). Data Structures and Algorithms Analysis in Java(4th ed.). New Delhi: Wiley.
11. Herbert Schildt. (2014). Java The Complete Reference (English)(9th ed.). New Delhi: Tata McGraw Hill.
12. Malik, D. S., & Nair, P.S. (2003).Data Structures Using Java. New Delhi: Course Technology.

### WEB SITES

[http://en.wikipedia.org/wiki/Data\\_structure](http://en.wikipedia.org/wiki/Data_structure)  
<http://www.cs.sunysb.edu/~skiena/214/lectures/>  
[www.amazon.com/Teach-Yourself-Structures-Algorithms](http://www.amazon.com/Teach-Yourself-Structures-Algorithms)

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

**Course Objectives**

- To Study the basic concepts and functions of operating systems.
- To understand the structure and functions of OS.
- To Learn about Processes, Threads and Scheduling algorithms.
- To Understand the principles of concurrency, Deadlocks and Memory Management
- To Learn about the Protection and Security Concepts.
- To provide experience on MS Windows and LINUX environment.

**Course Outcomes (COs)**

Upon completion of this course the students will be able to:

1. Design various Scheduling algorithms.
2. Apply the principles of concurrency.
3. Design deadlock, prevention and avoidance algorithms.
4. Compare and contrast various memory management schemes.
5. Apply the Security Concepts based on Authentication.
6. Work in MS Windows and LINUX environment.

**UNIT-I**

**Introduction to Operating System:** 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.

**UNIT-II**

**Operating System Organization:** Processor and user modes-Kernels-System Calls and System Programs. **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.

**UNIT-III**

**Memory Management:** Physical and Virtual address space-Memory Allocation strategies –Fixed and Variable partitions-Paging-Segmentation-Virtual memory.

**UNIT-IV**

**File and I/O Management:** Directory structure-File operations-File Allocation methods-Device management.

## **UNIT-V**

**Protection and Security:** Policy mechanism-Authentication-Internal aCSUess Authorization.

### **Suggested Readings**

1. Silberschatz, A ., Galvin, P.B. , & Gagne, G. (2008). Operating Systems Concepts, 8<sup>th</sup> ed.). New Delhi: John Wiley Publications.
2. Tanenbaum, A.S. (2007).Modern Operating Systems (3<sup>rd</sup> ed.). New Delhi: Pearson Education.
3. Stallings, W. ( 2008). Operating Systems, Internals & Design Principles (5th ed.). New Delhi: Prentice Hall of India.

### **WEB SITES**

1. [www.cs.columbia.edu/~nieh/teaching/e6118\\_s00/](http://www.cs.columbia.edu/~nieh/teaching/e6118_s00/)
2. [www.clarkson.edu/~jnm/cs644](http://www.clarkson.edu/~jnm/cs644)
3. [pages.cs.wisc.edu/~remzi/Classes/736/Fall2002/](http://pages.cs.wisc.edu/~remzi/Classes/736/Fall2002/)

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

### Course Objectives

- To master the fundamentals of data communications networks by gaining a working knowledge of data transmission concepts.
- To study the basic taxonomy and terminology of the computer networking and enumerate the layers of OSI model and TCP/IP model.
- To read the fundamentals and basics concepts of Physical layer with real time examples
- To study data link layer concepts, design issues, and protocols.
- To learn the functions of network layer and the various routing protocols.
- To familiarize the functions and protocols of the Transport layer and Application layer.

### Course Outcomes (COs)

Upon completion of this course the students will be able to:

1. Understand the functions of each layer in OSI and TCP/IP model.
2. Explain the multiplexing, switching concept and types of transmission media with real time examples.
3. Understand the error detection and correction methods and can implement the data link layer protocols
4. Understand channel error detection and correction, MAC protocols, Ethernet and WLAN.
5. Learn different medium access method to avoid collision and to learn about routing table.
6. Learn basic functionalities of transport layer and application layer.

### Unit I

**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. **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;

### Unit – II

( cont..)digital to analog modulation-; multiplexing techniques- FDM, TDM; transmission media.

**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.

### Unit – III

**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-n ARQ; Point to Point Protocol on Internet.

#### **Unit – IV**

**Multiple Access Protocol and Networks:** CSMA/CD protocols; Ethernet LANS; connecting LAN and back-bone networks- repeaters, hubs, switches, bridges, router and gateways; **Networks Layer Functions and Protocols:** Routing; routing algorithms; network layer protocol of Internet- IP protocol, Internet control protocols.

#### **Unit V**

**Transport Layer Functions and Protocols:** Transport services- error and flow control, Connection establishment and release- three way handshake; **Overview of Application layer protocol:** Overview of DNS protocol; overview of WWW &HTTP protocol.

#### **Suggested Readings**

1. Forouzan, B. A.( 2007). Data Communications and Networking(4<sup>th</sup> ed.). New Delhi: THM.
2. Tanenbaum, A. S. (2002). Computer Networks (4<sup>th</sup> ed.). New Delhi: PHI.

#### **WEB SITES**

1. [en.wikipedia.org/wiki/Internet\\_protocol\\_suite](http://en.wikipedia.org/wiki/Internet_protocol_suite)
2. [http://docwiki.cisco.com/wiki/Introduction\\_to\\_WAN\\_Technologies](http://docwiki.cisco.com/wiki/Introduction_to_WAN_Technologies)
3. [www.yale.edu/pclt/COMM/TCPIP.HTM](http://www.yale.edu/pclt/COMM/TCPIP.HTM)
4. [www.w3schools.com/tcpip/default.asp](http://www.w3schools.com/tcpip/default.asp)

---

Instruction Hours / week: L: 3 T: 0 P: 0    **Marks:** Int : 40 Ext : 60    Total: 100

**Course Objectives**

- To compare the differences between Android and other mobile development environments.
- To learn the Object-oriented features of Kotlin and APIs for Android Development.
- To describe the working of Android applications, life cycle, manifest, and Intents
- To demonstrate the implementation of Form widgets for Android App development.
- To learn the SQLite database connectivity and database operations with android
- To design, create, deploy, and test applications for the Android mobile phone platform.

**Course Outcomes (Cos)**

Upon completion of this course the students will be able to:

1. Design and develop useful Android applications with compelling user interfaces by using, extending, and creating your own layouts and Views and using Menus.
2. Analyze the Architecture and features of Android with another Mobile Operating System.
3. Evaluate the standard of Kotlin language for developing Android Applications
4. Apply knowledge for creating user Interface and develop activity for Android App.
5. Evaluate the user interface architecture of Android for developing Android Apps
6. Understand the implementation of SQLite database operations with Android.

**UNIT-I:**

**Introduction:** History of Android, Introduction to Android Operating Systems, Android Development Tools, Android Architecture.

**UNIT-II:**

**Overview of object oriented programming using Java:** OOPs Concepts: Inheritance, Polymorphism, Interfaces, Abstract class, Threads, Overloading and Overriding, Java Virtual Machine.

**UNIT-III:**

**Development Tools:** Installing and using Eclipse with ADT plug-in, Installing Virtual machine for Android sandwich/Jelly bean (Emulator), configuring the installed tools, creating a android project– Hello Word, run on emulator, Deploy it on USB-connected Android device.

## UNIT-IV

**User Interface Architecture:** Application context, intents, Activity life cycle, multiple screen sizes. **User Interface Design:** Form widgets, Text Fields, Layouts, Button control, toggle buttons, Spinners(Combo boxes), Images, Menu, Dialog.

## UNIT-V

**Database:** Understanding of SQLite database, connecting with the database.

### Suggested Readings

1. James, C. Sheusi.( 2013). Android application development for java for java programmers. New Delhi: Cengage Learning.

## REFERENCES

1. <http://www.developer.android.com>
2. <http://developer.android.com/about/versions/index.html>
3. <http://developer.android.com/training/basics/firstapp/index.html>
4. <http://docs.oracle.com/javase/tutorial/index.htm>(Available in the form of free downloadable ebooks also).
5. <http://developer.android.com/guide/components/activities.html>
6. <http://developer.android.com/guide/components/fundamentals.html>
7. <http://developer.android.com/guide/components/intents-filters.html>.
8. <http://developer.android.com/training/multiscreen/screensizes.html>
9. <http://developer.android.com/guide/topics/ui/controls.html>
10. <http://developer.android.com/guide/topics/ui/declaring-layout.html>
11. <http://developer.android.com/training/basics/data-storage/databases.html>

**Course Objectives**

- To understand the need for simulation/implementation for the verification of mathematical functions.
- To understand the main features of the MATLAB program development environment to enable their usage in the higher learning.
- To implement simple mathematical functions/equations in numerical computing environment such as MATLAB.
- To interpret and visualize simple mathematical functions and operations thereon using plots/display.
- To analyze the program for correctness and determine/estimate/predict the output and verify it under simulation environment using MATLAB tools.
- To apply MATLAB tools for implementation/simulation and visualization of basic mathematical functions

**Course Outcomes (Cos)**

On successful completion of the course, the students should be able to

1. Understand the need for simulation/implementation for the verification of mathematical functions.
2. Understand the main features of the MATLAB program development environment to enable their usage in the higher learning.
3. Implement simple mathematical functions/equations in numerical computing environment such as MATLAB.
4. Interpret and visualize simple mathematical functions and operations thereon using plots/display.
5. Analyze the program for correctness and determine/estimate/predict the output and verify it under simulation environment using MATLAB tools.
6. Apply MATLAB tools for implementation/simulation and visualization of basic mathematical functions

**UNIT-I**

**Introduction to Programming:** Components of a computer, working with numbers, Machine code, Software hierarchy.

**UNIT-II**

**Programming Environment:** MATLAB Windows, A First Program, Expressions, Constants, Variables and assignment statement, Arrays.



### **UNIT-III**

**Graph Plots:** Basic plotting, Built in functions, Generating waveforms, Sound replay, load and save. **Procedures and Functions:** Arguments and return values, M-files, Formatted console input-output , String handling,

### **UNIT-IV**

**Control Statements:** Conditional statements: If, Else, Else-if, Repetition statements: While, for loop

### **UNIT-V**

**Manipulating Text:** Writing to a text file, Reading from a text file, Randomising and sorting a list, searching a list. **GUI Interface:** Attaching buttons to actions, Getting Input, Setting Output

### **Suggested Readings**

1. Amos Gilat. MATLAB: An Introduction with Applications(2nd ed). New Delhi: Wiley.
2. Moler, C.B. (2004). Numerical Computing with MATLAB. New Delhi: SIAM.

---

**Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60****Total: 100****Course Objectives**

- To understand the fundamental concepts of data structures
- To Learn linear data structures – lists, stacks, and queues
- To apply Tree and Graph structures
- To understand and apply sorting, searching algorithms
- To know about hashing algorithms
- To develop application using data structures

**Course Outcomes (COs)**

Upon completion of this course, the student will be able to:

1. Implement abstract data types for linear data structures.
2. Apply the different linear and non-linear data structures to problem solutions.
3. Analyze the applications of tree.
4. Implement graph theory over various data structures.
5. Critically analyze the various sorting algorithms.
6. Apply searching algorithms over various data structures.

**List of programs**

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search. Use Template functions.
2. WAP using templates to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.
3. Implement Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists (include a function and also overload operator +).
4. Implement Doubly Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
5. Implement Circular Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
6. Perform Stack operations using Linked List implementation.
7. Perform Stack operations using Array implementation. Use Templates.
8. Perform Queues operations using Circular Array implementation. Use Templates.
9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. WAP to scan a polynomial using linked list and add two polynomial.
11. WAP to calculate factorial and to compute the factors of a given no. (i)using recursion, (ii) using iteration
12. (ii) WAP to display Fibonacci series (i)using recursion, (ii) using iteration
13. WAP to calculate GCD of 2 number (i) with recursion (ii) without recursion

14. WAP to create a Binary Search Tree and include following operations in tree: (a) Insertion (Recursive and Iterative Implementation)
  - (b) Deletion by copying
  - (c) Deletion by Merging
  - (d) Search a no. in BST
  - (e) Display its preorder, postorder and inorder traversals Recursively
  - (f) Display its preorder, postorder and inorder traversals Iteratively
  - (g) Display its level-by-level traversals
  - (h) Count the non-leaf nodes and leaf nodes
  - (i) Display height of tree
  - (j) Create a mirror image of tree
  - (k) Check whether two BSTs are equal or not
15. WAP to convert the Sparse Matrix into non-zero form and vice-versa.
16. WAP to reverse the order of the elements in the stack using additional stack.
17. WAP to reverse the order of the elements in the stack using additional Queue.
18. WAP to implement Diagonal Matrix using one-dimensional array.
19. WAP to implement Lower Triangular Matrix using one-dimensional array.
20. WAP to implement Upper Triangular Matrix using one-dimensional array.
21. WAP to implement Symmetric Matrix using one-dimensional array.
22. WAP to create a Threaded Binary Tree as per inorder traversal, and implement operations like finding the successor / predecessor of an element, insert an element, inorder traversal.
23. WAP to implement various operations on AVL Tree.

---

**Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60****Total: 100****Course Objectives**

- To Study the basic concepts and functions of operating systems.
- To understand the structure and functions of OS.
- To Learn about Processes, Threads and Scheduling algorithms.
- To Understand the principles of concurrency, Deadlocks and Memory Management
- To Learn about the Protection and Security Concepts.
- To provide experience on MS Windows and LINUX environment.

**Course Outcomes (COs)**

Upon completion of this course the students will be able to:

1. Design various Scheduling algorithms.
2. Apply the principles of concurrency.
3. Design deadlock, prevention and avoidance algorithms.
4. Compare and contrast various memory management schemes.
5. Apply the Security Concepts based on Authentication.
6. Work in MS Windows and LINUX environment.

**List of programs**

1. Write a program (using *fork()* and/or *exec()* commands) where parent and child execute:
  - 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 program to implement FCFS scheduling algorithm.
7. Write program to implement Round Robin scheduling algorithm.
8. Write program to implement SJF scheduling algorithm.
9. Write program to implement non-preemptive priority based scheduling algorithm.
10. Write program to implement preemptive priority based scheduling algorithm.
11. Write program to implement SRJF scheduling algorithm.
12. Write program to calculate sum of n numbers using *thread* library.
13. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.

---

**Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60 Total: 100**

**Course Objectives**

- To master the fundamentals of data communications networks by gaining a working knowledge of data transmission concepts.
- To study the basic taxonomy and terminology of the computer networking and enumerate the layers of OSI model and TCP/IP model.
- To read the fundamentals and basics concepts of Physical layer with real time examples
- To study data link layer concepts, design issues, and protocols.
- To learn the functions of network layer and the various routing protocols.
- To familiarize the functions and protocols of the Transport layer and Application layer.

**Course Outcomes (COs)**

Upon completion of this course the students will be able to:

1. Understand the functions of each layer in OSI and TCP/IP model.
2. Explain the multiplexing, switching concept and types of transmission media with real time examples.
3. Understand the error detection and correction methods and can implement the data link layer protocols
4. Understand channel error detection and correction, MAC protocols, Ethernet and WLAN.
5. Learn different medium access method to avoid collision and to learn about routing table.
6. Learn basic functionalities of transport layer and application layer.

**List of Programs**

1. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.
2. Simulate and implement stop and wait protocol for noisy channel.
3. Simulate and implement go back n 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.

---

**Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Int : 40 Ext : 60****Total: 100****Course Objectives**

- To compare the differences between Android and other mobile development environments.
- To learn the Object-oriented features of Kotlin and APIs for Android Development.
- To describe the working of Android applications, life cycle, manifest, and Intents
- To demonstrate the implementation of Form widgets for Android App development.
- To learn the SQLite database connectivity and database operations with android
- To design, create, deploy, and test applications for the Android mobile phone platform.

**Course Outcomes (Cos)**

Upon completion of this course the students will be able to:

1. Design and develop useful Android applications with compelling user interfaces by using, extending, and creating your own layouts and Views and using Menus.
2. Analyze the Architecture and features of Android with another Mobile Operating System.
3. Evaluate the standard of Kotlin language for developing Android Applications
4. Apply knowledge for creating user Interface and develop activity for Android App.
5. Evaluate the user interface architecture of Android for developing Android Apps
6. Understand the implementation of SQLite database operations with Android.

**List of Programs**

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 login module. (Check username and password).
3. Create spinner with strings taken from resource folder (res >> value folder) and on changing the spinner value, Image will change.
4. Create a menu with 5 options and selected option should appear in 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.

---

**Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Int : 40 Ext : 60****Total: 100****Course Objectives**

- To understand the need for simulation/implementation for the verification of mathematical functions.
- To understand the main features of the MATLAB program development environment to enable their usage in the higher learning.
- To implement simple mathematical functions/equations in numerical computing environment such as MATLAB.
- To interpret and visualize simple mathematical functions and operations thereon using plots/display.
- To analyze the program for correctness and determine/estimate/predict the output and verify it under simulation environment using MATLAB tools.
- To apply MATLAB tools for implementation/simulation and visualization of basic mathematical functions

**Course Outcomes (Cos)**

On successful completion of the course, the students should be able to

1. Understand the need for simulation/implementation for the verification of mathematical functions.
2. Understand the main features of the MATLAB program development environment to enable their usage in the higher learning.
3. Implement simple mathematical functions/equations in numerical computing environment such as MATLAB.
4. Interpret and visualize simple mathematical functions and operations thereon using plots/display.
5. Analyze the program for correctness and determine/estimate/predict the output and verify it under simulation environment using MATLAB tools.
6. Apply MATLAB tools for implementation/simulation and visualization of basic mathematical functions

**List of programs**

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. 2, 4, 6, 8, 10

b. 1/2, 1, 3/2, 2, 5/2

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 co-ordinates :

$$r_n = \sqrt{n}$$

$$\theta_n = \frac{137.51}{180} \pi n$$

10. Calculate 10 approximate points from the function  $y=2x$  by using the formulae:

i.  $x_n = n$

ii.  $y_n = 2n + \text{rand} - 0.5$

Fit a line of best fit to these points using the function polyfit() with degree=1, and generate co-ordinates 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 sinewave at 500Hz with a sampling rate of 11025Hz. 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.

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:

- i. FtoC(96)
- ii. lookfor Fahrenheit
- iii. help FtoC

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:

- i. Enter string 1: Mark
- ii. Enter string 2: Huckvale
- iii. Mark Huckvale
- iv. \*\*\*\*\*
- v. elavkcuH kraM

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

**Course Objectives**

- To learn mathematical background for analysis of algorithm
- To learn various advanced data structures.
- To understand the concept of designing an algorithm.
- To learn dynamic programming and greedy method.
- To understand the concept of pattern matching
- To learn advanced tree and graph applications

**Course Outcomes(COs)**

1. Learn to choose appropriate advanced data structure for given problem
2. Knowledge to calculate complexity.
3. Select appropriate design techniques to solve real world problems.
4. Apply the dynamic programming technique to solve the problems.
5. Apply the greedy programming technique to solve the problems.
6. Select a proper pattern matching algorithm for given problem

**UNIT-I**

**Introduction:** Basic Design and Analysis techniques of Algorithms, Correctness of Algorithm.

**Algorithm Design Techniques:** Iterative techniques, Divide and Conquer, Dynamic Programming, Greedy Algorithms.

**UNIT-II**

**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;

**UNIT-III**

**Lower Bounding Techniques:** Decision Trees **Balanced Trees:** Red-Black Trees

**UNIT-IV**

**Advanced Analysis Technique:** Amortized analysis **Graphs:** Graph Algorithms– Breadth First Search, Depth First Search and its Applications, Minimum Spanning Trees.

**UNIT-V**

**String Processing:** String Matching, KMP Technique.

## **Suggested Readings**

1. Cormen, T.H., Charles, E. Leiserson., Ronald, L. Rivest. (2009). Clifford Stein Introduction to Algorithms(3rd ed.). New Delhi: PHI.
2. Sarabasse., Gelder, A.V. (1999). Computer Algorithm – Introduction to Design and Analysis (3rd ed.). New Delhi: Pearson

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

### Course Objectives

- To Apply their knowledge of mathematics, sciences, and computer science to the modeling, analysis, and measurement of software artifacts.
- To Work effectively as leader/member of a development team to deliver quality software artifacts.
- To Analyze, specify and document software requirements for a software system.
- To Implement a given software design using sound development practices.
- To Verify, validate, assess and assure the quality of software artifacts.
- To Design, select and apply the most appropriate software engineering process for a given project, plan for a software project, identify its scope and risks, and estimate its cost and time.

### Course Outcomes(COs)

1. Identify suitable life cycle models to be used and translate a requirement specification to a design using an appropriate software engineering methodology.
2. Apply systematic procedure for software design and deployment.
3. Analyze a problem and identify and define the computing requirements to the problem.
4. Formulate appropriate testing strategy for the given software system.
5. Develop software projects based on current technology, and test the software using testing tools.
6. Design, select and apply the most appropriate software engineering process for a given project, plan for a software project, identify its scope and risks, and estimate its cost and time.

### UNIT-I

**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).

### UNIT-II:

**Requirement Analysis;** Initiating Requirement Engineering Process- Requirement Analysis and Modeling Techniques- Flow Oriented Modeling- Need for SRS- Characteristics and Components of SRS- Software Project Management: Estimation in Project Planning Process, Project Scheduling.

### UNIT-III:

Risk Management: Software Risks, Risk Identification Risk Projection and Risk Refinement, RMMM plan, **Quality Management-** Quality Concepts, Software Quality Assurance, Software Reviews, Metrics for Process and Projects

#### **UNIT-IV:**

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

#### **UNIT-V**

**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

#### **Suggested Readings**

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

#### **WEB SITES**

1. [http://en.wikipedia.org/wiki/Software\\_engineering](http://en.wikipedia.org/wiki/Software_engineering)
2. <http://www.onesmartclick.com/engineering/software-engineering.html>
3. [http://www.CSU.gatech.edu/classes/AY2000/cs3802\\_fall/](http://www.CSU.gatech.edu/classes/AY2000/cs3802_fall/)

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

### Course Objectives

- To describe a good introduction to the discipline of database management systems.
- To give a good formal foundation on the data models and E-R model.
- To demonstrate the principles database constraints behind systematic database design by covering normalization concept.
- To introduce the concepts of basic SQL as a universal Database language.
- To retrieve any type of information from a data base by formulating complex queries in SQL.
- To analyze the existing design of a database schema and apply concepts of normalization to design an optimal database.

### Course Outcomes (COs)

Upon completion of this course, students will be able to:

1. Demonstrate an understanding of the elementary features of RDBMS
2. Design conceptual models of a database using ER modeling for real life applications
3. Analyze the existing design of a database schema and apply concepts of normalization to design an optimal database
4. Able to develop structured query language (SQL) queries to create, read, update, and delete relational database
5. Retrieve any type of information from a data base by formulating complex queries in SQL.
6. Analyze the existing design of a database schema and apply concepts of normalization to design an optimal database.

### UNIT-I

**Introduction:** Characteristics of database approach, data models, database system architecture and data independence. **Entity Relationship(ER) Modeling:** Entity types, relationships, constraints.

### UNIT-II

**Relation data model:** Relational model concepts, relational constraints, relational algebra.

### UNIT-III

**Relation data model:** SQLqueries **Database design:** Mapping ER/EER model to relational database, functional dependencies, Lossless decomposition.

### UNIT-IV

**Database design:** Normal forms (upto BCNF).**Transaction Processing :** ACID properties, concurrency control

#### **UNIT-V**

**File Structure and Indexing** (8 Lectures) 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.

#### **References:**

1. Elmasri, R., & Navathe, S.B. (2010). Fundamentals of Database Systems (6th ed.). New Delhi: Pearson Education,.
2. Ramakrishanan, R., & Gehrke, J. (2002). Database Management Systems (3rd ed.). New Delhi: McGraw-Hill.
3. Silberschatz, A., Korth, H.F., & Sudarshan, S. (2010). Database System Concepts (6th ed.). New Delhi: McGraw-Hill
4. Elmasri, R., & Navathe, S.B. (2013). Database Systems Models, Languages, Design and application Programming (6th ed.). New Delhi: Pearson Education.

#### **WEB SITES**

1. <http://en.wikipedia.org/wiki/RDBMS>
2. [http://aspalliance.com/1211\\_Relational\\_Database\\_Management\\_Systems\\_\\_Concepts\\_and\\_Terminologies](http://aspalliance.com/1211_Relational_Database_Management_Systems__Concepts_and_Terminologies)
3. [www.compinfo-center.com/apps/rdbms.html](http://www.compinfo-center.com/apps/rdbms.html)

---

Instruction Hours / week: L: 3 T: 0 P: 0    **Marks:** Int : 40 Ext : 60    Total: 100

**Course Objectives**

- To Understand the fundamentals of HTML and use different formatting options
- To Create tables and frames
- To insert a graphic and links within a web page.
- To insert ordered and unordered lists within a web page.
- To use cascading style sheets.
- To validate and publish a web page.

**Course Outcomes (COs)**

1. Understand the fundamentals of HTML and use different formatting options
2. Create tables and frames
3. Insert a graphic and links within a web page.
4. Insert ordered and unordered lists within a web page.
5. Use cascading style sheets.
6. Validate and publish a web page.

**Unit-I:**

**Introduction The Basics: The Head, the Body, Colors, Attributes, Lists, ordered and unordered**

**Unit-II**

**Links Introduction: Relative Links, Absolute Links ,Link Attributes ,Using the ID Attribute to Link Within a Document**

**Unit-III:**

**Images: Putting an Image on a Page ,Using Images as Links, Putting an Image in the Background**

**Unit IV:**

**Tables: Creating a Table, Table Headers, Captions, Spanning Multiple Columns ,Styling Table**

**Unit V :**

**Forms :Basic Input and Attributes ,Other Kinds of Inputs ,Styling forms with CSS ,Where To Go From Here**

**Suggested Readings**

1. Virginia DeBolt. (2006). Integrated HTML and CSS A Smarter, Faster Way to Learn New Delhi: Wiley / Sybex.



2. Cassidy Williams., & Camryn Williams. (2015). Introduction to **HTML** and **CSS**, O'Reilly.

### **WEB SITES**

1. [www.w3schools.com/](http://www.w3schools.com/)
2. [alexle.net/archives/category/web-technolgy](http://alexle.net/archives/category/web-technolgy)
3. [jmarshall.com/easy/](http://jmarshall.com/easy/)
4. [www.php.net/](http://www.php.net/)
5. [en.wikipedia.org/wiki/PHP](http://en.wikipedia.org/wiki/PHP)
6. [www.w3schools.com/PHP/DEfaULT.asP](http://www.w3schools.com/PHP/DEfaULT.asP)

---

Instruction Hours / week: L: 3 T: 0 P: 0    **Marks:** Int : 40 Ext : 60    Total: 100

**Course Objective**

- To learn rules and techniques to create well-formed XML documents, learning to use XML namespaces correctly.
- To Construct Document Type Definitions and XML Schema documents that can be used to validate XML documents (structure, content).
- To develop dynamic web pages using XSL, applying XSLT transformations and formatting to XML documents (XSL, XPath).
- To create valid HTML webpages and Cascading Style Sheets, based on the specifications of W3C.
- To learn to display XML documents using CSS.
- To construct of complex queries over XML documents using XPath and XQuery.

**Course Outcomes(COs)**

1. Learning rules and techniques to create well-formed XML documents, learning to use XML namespaces correctly.
2. Constructing Document Type Definitions and XML Schema documents that can be used to validate XML documents (structure, content).
3. Developing dynamic web pages using XSL, applying XSLT transformations and formatting to XML documents (XSL, XPath).
4. Creating valid HTML webpages and Cascading Style Sheets, based on the specifications of W3C.
5. Learning to display XML documents using CSS.
6. Construction of complex queries over XML documents using XPath and XQuery.

**UNIT-I**

**Introduction:** Understanding Mark-up Languages, Introduction to XML and its Goals.

**UNIT-II**

**XML Basics:** XML Structure and Syntax, Document classes and Rules.

**UNIT-III**

**Other XML Concepts:** Scripting XML

**UNIT-IV**

**Other XML Concepts:** XML as Data, Linking with XML

**UNIT-V**

**XML with Style:** XSL –Style Sheet Basics, XSL basics, XSL style sheets.

## **Suggested Readings**

1. William, J. Pardi. XML in action web technology.
2. Michael, J. Young. Step by Step XML.

---

**Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60**

**Total: 100**

**Course Objectives**

- To learn mathematical background for analysis of algorithm
- To learn various advanced data structures.
- To understand the concept of designing an algorithm.
- To learn dynamic programming and greedy method.
- To understand the concept of pattern matching
- To learn advanced tree and graph applications

**Course Outcomes(COs)**

1. Learn to choose appropriate advanced data structure for given problem
2. Knowledge to calculate complexity.
3. Select appropriate design techniques to solve real world problems.
4. Apply the dynamic programming technique to solve the problems.
5. Apply the greedy programming technique to solve the problems.
6. Select a proper pattern matching algorithm for given problem

**List of Programs**

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

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

**Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60****Total: 100****Course Objectives**

- To Apply their knowledge of mathematics, sciences, and computer science to the modeling, analysis, and measurement of software artifacts.
- To Work effectively as leader/member of a development team to deliver quality software artifacts.
- To Analyze, specify and document software requirements for a software system.
- To Implement a given software design using sound development practices.
- To Verify, validate, assess and assure the quality of software artifacts.
- To Design, select and apply the most appropriate software engineering process for a given project, plan for a software project, identify its scope and risks, and estimate its cost and time.

**Course Outcomes(COs)**

1. Identify suitable life cycle models to be used and translate a requirement specification to a design using an appropriate software engineering methodology.
2. Apply systematic procedure for software design and deployment.
3. Analyze a problem and identify and define the computing requirements to the problem.
4. Formulate appropriate testing strategy for the given software system.
5. Develop software projects based on current technology, and test the software using testing tools.
6. Design, select and apply the most appropriate software engineering process for a given project, plan for a software project, identify its scope and risks, and estimate its cost and time.

S. No	Practical Title
1.	<ul style="list-style-type: none"> <li>• Problem Statement</li> <li>• Process Model</li> </ul>
2.	Requirement Analysis: <ul style="list-style-type: none"> <li>• Creating a Data Flow</li> <li>• Data Dictionary, Use Cases</li> </ul>
3.	Project Management: <ul style="list-style-type: none"> <li>• Computing FP</li> <li>• Effort</li> <li>• Schedule, Risk Table, Timeline chart</li> </ul>
4.	Design Engineering: <ul style="list-style-type: none"> <li>• Architectural Design</li> <li>• Data Design, Component Level Design</li> </ul>
5.	Testing: <ul style="list-style-type: none"> <li>• Basis Path Testing</li> </ul>

**Sample Projects:**

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

**Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60 Total: 100**

**Course Objectives**

- To describe a good introduction to the discipline of database management systems.
- To give a good formal foundation on the data models and E-R model.
- To demonstrate the principles database constraints behind systematic database design by covering normalization concept.
- To introduce the concepts of basic SQL as a universal Database language.
- To retrieve any type of information from a data base by formulating complex queries in SQL.
- To analyze the existing design of a database schema and apply concepts of normalization to design an optimal database.

**Course Outcomes (COs)**

Upon completion of this course, students will be able to:

1. Demonstrate an understanding of the elementary features of RDBMS
2. Design conceptual models of a database using ER modeling for real life applications
3. Analyze the existing design of a database schema and apply concepts of normalization to design an optimal database
4. Able to develop structured query language (SQL) queries to create, read, update, and delete relational database
5. Retrieve any type of information from a data base by formulating complex queries in SQL.
6. Analyze the existing design of a database schema and apply concepts of normalization to design an optimal database.

1. Create and use the following database schema to answer the given queries.

**EMPLOYEE Schema**

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

**DEPARTMENT Schema**

Field	Type	NULL	KEY	DEFAULT
Dno	Integer	No	PRI	NULL
Dname	Varchar(50)	Yes		NULL
Location	Varchar(50)	Yes		New Delhi

### Query List

1. Query to display Employee Name, Job, Hire Date, Employee Number; for each employee with the Employee Number appearing first.
2. Query to display unique Jobs from the Employee Table.
3. Query to display the Employee Name concatenated by a Job separated by a comma.
4. Query to display all the data from the Employee Table. Separate each Column by a comma and name the said column as THE\_OUTPUT.
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 Name of all the employees where the third letter of their name is A.
14. Query to display 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 Manger'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 Current Date.
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 letter 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 Manger'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 asterisk (\*) signifies \$100.
29. Query to display the Highest, Lowest, Sum and Average Salaries of all the employees
30. Query to display the number of employees performing the same Job type functions.
31. Query to display the no. of managers without listing their names.
32. Query to display the Department Name, Location Name, No. of Employees and the average salary for all employees in that department.
33. Query to display Name and Hire Date for all employees in the same dept. as Blake.
34. Query to display the Employee No. And Name for all employees who earn more than the average salary.
35. Query to display Employee Number and Name for all employees who work in a department with any employee whose name contains a '\_T'.
36. Query to display the names and salaries of all employees who report to King.
37. Query to display the department no, name and job for all employees in the Sales department.

---

**Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Int : 40 Ext : 60****Total: 100****Course Objectives**

- To Understand the fundamentals of HTML and use different formatting options
- To Create tables and frames
- To insert a graphic and links within a web page.
- To insert ordered and unordered lists within a web page.
- To use cascading style sheets.
- To validate and publish a web page.

**Course Outcomes (COs)**

1. Understand the fundamentals of HTML and use different formatting options
2. Create tables and frames
3. Insert a graphic and links within a web page.
4. Insert ordered and unordered lists within a web page.
5. Use cascading style sheets.
6. Validate and publish a web page.

**List of Programs**

1. Create an HTML document with the following formatting options:

I. Bold

II. Italics

III. Underline

IV. Headings (Using H1 to H6 heading styles)

V. Font (Type, Size and Color)

VI. Background (Colored background/Image in background)

VII. Paragraph

VIII. Line Break

IX. Horizontal Rule

X. Pre tag

2. Create an HTML document which consists of:

I. Ordered List

II. Unordered List

III. Nested List

IV. 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

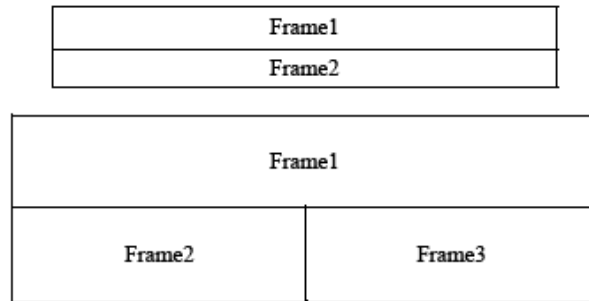
5 Create a Table with the following view:

				Place an image here

6 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

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



---

**Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Int : 40 Ext : 60****Total: 100****Course Objective**

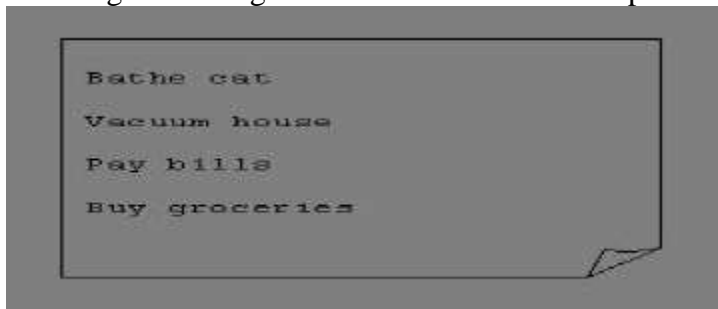
- To learn rules and techniques to create well-formed XML documents, learning to use XML namespaces correctly.
- To Construct Document Type Definitions and XML Schema documents that can be used to validate XML documents (structure, content).
- To develop dynamic web pages using XSL, applying XSLT transformations and formatting to XML documents (XSL, XPath).
- To create valid HTML webpages and Cascading Style Sheets, based on the specifications of W3C.
- To learn to display XML documents using CSS.
- To construct of complex queries over XML documents using XPath and XQuery.

**Course Outcomes(COs)**

1. Learning rules and techniques to create well-formed XML documents, learning to use XML namespaces correctly.
2. Constructing Document Type Definitions and XML Schema documents that can be used to validate XML documents (structure, content).
3. Developing dynamic web pages using XSL, applying XSLT transformations and formatting to XML documents (XSL, XPath).
4. Creating valid HTML webpages and Cascading Style Sheets, based on the specifications of W3C.
5. Learning to display XML documents using CSS.
6. Construction of complex queries over XML documents using XPath and XQuery.

**List of Programs****Exercise #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. 1. Draw a tree representation of the structure.

**Exercise 2# Deconstructing an XML Document**

In this exercise, student 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>
```

### **Exercise #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.

### **Exercise #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>
```

### **Exercise #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 warm water</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>
```

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

### Course Objectives

- To provide an overview of Information Security and Assurance.
- To provide an exposure to the spectrum of security activities methods methodologies and procedures with emphasis on practical aspects of Information Security.
- To explain concepts related to applied cryptography including the techniques for crypto-analysis symmetric and asymmetric cryptography.
- To understand the concepts of digital signature, message authentication code, hash functions and modes of encryption operations.
- To explain common vulnerabilities in computer programs including buffer overflow Vulnerabilities time-of-check to time-of-use flaws incomplete mediation.
- To understand the concepts of cryptographic utilities and authentication mechanisms to design secure applications

### Course Outcomes(COs)

A student who successfully completes this course should at a minimum be able to:

1. Explore the working principles and utilities of various cryptographic algorithms including secret key cryptography, hashes and message digests, and public key algorithms
2. State the basic concepts in information security
3. Explain concepts related to applied cryptography including the techniques for crypto-analysis symmetric and asymmetric cryptography.
4. Use digital signature, message authentication code, hash functions and modes of encryption operations.
5. Explain common vulnerabilities in computer programs including buffer overflow Vulnerabilities time-of-check to time-of-use flaws incomplete mediation.
6. Understand the concepts of cryptographic utilities and authentication mechanisms to design secure applications

### Unit I

**Introduction :** Security, Attacks, Computer Criminals, Security Services, Security Mechanisms.

**Cryptography :** Substitution ciphers, Transpositions Cipher, Confusion, diffusion.

### Unit – II

Symmetric, Asymmetric Encryption. DES Modes of DES, Uses of Encryption, Hash function, key exchange, Digital Signatures, Digital Certificates

### Unit – III

**Program Security:** Secure programs, Non malicious Program errors, Malicious codes virus, Trap doors, Salami attacks, Covert channels, Control against program. **Threats:**

Protection in OS: Memory and Address Protection, ACSUess control, File Protection, User Authentication.

#### **Unit – IV**

**Database Security:** Requirements, Reliability, Integrity, Sensitive data, Inference, Multilevel Security. **Security in Networks :** Threats in Networks, Security Controls, firewalls, Intrusion detection systems, Secure e-mails

#### **Unit V**

##### **Administrating Security**

Security Planning, Risk Analysis, Organisational Security Policy, Physical Security. Ethical issues in Security: Protecting Programs and data. Information and law.

##### **Suggested Readings**

1. Pfleeger, C. P., & Pfleeger, S. L.( 2006). Security in Computing. New Delhi: Prentice Hall of India
2. Stallings, W. (2010). Network Security Essentials: Applications and Standards(4<sup>th</sup> ed.).

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

**Course Objectives**

- To describe the fundamental concepts of networks
- To demonstrate mastery of main protocols comprising the Internet.
- To develop skills in network programming techniques.
- To implement network services that communicate through the Internet.
- To apply the client - server model in networking applications.
- To practice networking commands available through the operating systems

**Course Outcomes(COs)**

Having successfully completed this course, the student will be able to:

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

**UNIT-I**

Transport Layer Protocols: TCP, UDP, SCTP protocol.

**UNIT-II**

Socket Programming: Socket Introduction; TCP Sockets; TCP Client/Server Example ; signal handling

**UNIT-III**

I/O multiplexing using sockets; Socket Options; UDP Sockets; UDP client server example; Address lookup using sockets.

**UNIT-IV**

Network Applications: Remote logging; Email; WWW and HTTP.

**UNIT-V**

LAN administration: Linux and TCP/IP networking: Network Management and Debugging.

**Suggested Readings**

1. Richard Stevens, W., Bill Fenner., & Andrew, M. Rudoff. (2003). Unix Network Programming, The sockets Networking API, Vol. 1(3rd ed.). New Delhi: PHI.



2. Forouzan, B. A. (2003). Data Communications and Networking(4th ed.). New Delhi: THM Publishing Company Ltd.,
3. Nemeth Synder., & Hein. (2010). Linux Administration Handbook (2nd ed.), New Delhi: Pearson Education.
4. Steven, R. (1990). Unix Network Programming (2nd ed.). New Delhi: PHI.

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

**Course Objectives**

- To apply the fundamentals of assembly level programming of microprocessors.
- To build a program on a microprocessor using arithmetic & logical instruction set of 8086.
- To develop the assembly level programming using 8086 loop instruction set.
- To write programs based on string and procedure for 8086 microprocessor.
- To analyze abstract problems and apply a combination of hardware and software to address the problem
- To make use of standard test and measurement equipment to evaluate digital interfaces

**Course Outcomes(COs)**

1. Apply the fundamentals of assembly level programming of microprocessors.
2. Build a program on a microprocessor using arithmetic & logical instruction set of 8086.
3. Develop the assembly level programming using 8086 loop instruction set.
4. Write programs based on string and procedure for 8086 microprocessor.
5. Analyze abstract problems and apply a combination of hardware and software to address the problem
6. Make use of standard test and measurement equipment to evaluate digital interfaces

**Unit – I- Introduction to Microprocessor**

Introduction to 8085 – Pin Diagram – Microprocessor architecture and its operations– Demultiplexing the Bus – Generation of Control Signals – Fetching, Decoding and Execution of Instruction – Instruction Timing and Status Flag.

**Unit – II – Addressing Modes**

Instruction Set – Addressing Modes – Instruction Format – Simple Program – Memory and Machine Cycle – Memory Units Machine Cycle.

**Unit – III – Interfacing Concepts**

Peripheral I/O Instructions – Device Selection and Data Transfer – Types of Data Transfer – Input Interfacing – Input Interfacing using Decoders – Output Interfacing – LED and 7 Segment Display – Interfacing Memory.

**Unit – IV - Peripheral Devices**

Introduction to Programmable Peripheral Interface 8255 – Pin Diagram – Architecture – Modes of Operation: I/O and BSR – Architecture and Operation of 8251 (USART) – Architecture and Operation of Programmable Interrupt Controller (8259) – Architecture of 254(8253) Programmable Interval Timer/ Counter – DMA Controller(8259).

## **Unit – V - Applications**

Delay Program – Traffic Light Control System – Water Level Controller – Stepper Motor Control – Interfacing DAC – Interfacing ADC – Temperature Measurement.

### **Suggested Readings:**

1. Barry, B. Brey. (2009). The Intel Microprocessors: Architecture, Programming and Interfacing (6th ed.). New Delhi: Pearson Education,
2. Gupta, M.K. (2006). Microprocessor Microcomputer, Microcontroller and Interfacing (First ed.). New Delhi: Paragon International Publisher.
3. Walter, A Triebel., & Avtar Singh.( 2005). The 8088 and 8086 Microprocessors Programming, Interfacing, Software, Hardware, and Applications (4th ed.). New Delhi: PHI.
4. Ramesh, S.Gaonkar. (2000). Microprocessor Architecture, Programming and Application with 8085(4th ed.). New Delhi: Penram International Publishing.

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

### Course Objectives

- To understand the fundamental concepts of a digital image processing system.
- To understand concepts of image enhancement techniques.
- To apply various Image Transforms.
- To analyze various Compression techniques and Morphological concepts
- To know various segmentation techniques, and object descriptors.
- To apply color models and various applications of image processing

### Course Outcomes (COs)

1. Remember the fundamental concepts of image processing.
2. Explain different Image enhancement techniques
3. Understand and review image transforms
4. Analyze the basic algorithms used for image processing & image compression with morphological image processing.
5. Contrast Image Segmentation and Representation
6. Design & Synthesize Color image processing and its real world applications

### UNIT-I

Introduction: Light, Brightness adaption and discrimination, Pixels, coordinate conventions, Imaging Geometry, Perspective Projection, Spatial Domain Filtering, sampling and quantization. Spatial Domain Filtering: Intensity transformations, contrast stretching, histogram equalization, Correlation and convolution, Smoothing filters, sharpening filters, gradient and Laplacian.

### UNIT-II

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.

### UNIT-III:

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, 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.

## **UNIT – IV**

FAX compression (CSUITT 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

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

## **UNIT-V**

Morphological Image Processing: Basics, SE, Erosion, Dilation, Opening, Closing, Hit-or-Miss Transform, Boundary Detection, Hole filling, Connected components, convex hull, thinning, thickening, skeletons, pruning, Geodesic Dilation, Erosion, Reconstruction by dilation and erosion. 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

### **Suggested Readings**

1. Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing(3rd ed.). New Delhi: Pearson Education.
2. Jain, A. K. (1989). Fundamentals of Digital image Processing. New Delhi: Prentice Hall of India.
3. Castleman, K. R. (1996). Digital Image Processing. New Delhi: Pearson Education.
4. Schalkoff. (1989). Digital Image Processing and Computer Vision. New York: John Wiley and Sons.
5. Rafael, C. Gonzalez., Richard, E. Woods.,& Steven Eddins. (2004). Digital Image Processing using MATLAB. New Delhi: Pearson Education.

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

### Course Objectives

- To introduce students to the basic concepts and techniques of Machine Learning.
- To develop skills of using recent machine learning software for solving practical problems.
- To gain experience of doing independent study and research.
- To recognize the characteristics of machine learning that make it useful to real-world problems.
- To characterize machine learning algorithms as supervised, semi-supervised, and unsupervised.
- To effectively use machine learning toolboxes.

### Course Outcomes(COs)

On successful completion of the course the student should be

1. Remember the basic concepts and techniques of Machine Learning.
2. Develop skills of using recent machine learning software for solving practical problems.
3. of doing independent study and research.
4. To recognize the characteristics of machine learning that make it useful to real-world problems.
5. To characterize machine learning algorithms as supervised, semi-supervised, and unsupervised.
6. To effectively use machine learning toolboxes.

### UNIT-I

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

### UNIT-II

Softwares for Machine Learning and Linear Algebra Overview : Plotting of Data, Vectorization, Matrices and Vectors: Addition, Multiplication, Transpose and Inverse using available tool such as MATLAB.

### UNIT-III

Linear Regression: Prediction using Linear Regression, Gradient Descent, Linear Regression with one variable, Linear Regression with multiple variables, Polynomial Regression, Feature Scaling/Selection.

Logistic Regression: Classification using Logistic Regression, Logistic Regression vs. Linear Regression, Logistic Regression with one variable and with multiple variables.

#### **UNIT-IV**

Regularization: Regularization and its utility: The problem of Overfitting, Application of Regularization in Linear and Logistic Regression, Regularization and Bias/Variance.

#### **UNIT-V**

Neural Networks: Introduction, Model Representation, Gradient Descent vs. Perceptron Training, Stochastic Gradient Descent, Multilayer Perceptrons, Multiclass Representation, Backpropagation Algorithm.

#### **Suggested Readings**

1. Santanu Chattopadhyaya. (2011). Systems Programming. New Delhi: PHI.
2. Alfred, V. Aho., Monica, S. Lam., Ravi Sethi., & Jeffrey, D. Ullman. (2006). Compilers: Principles, Techniques, and Tools (2nd ed.). New Delhi: Prentice Hall.
3. Dhamdhare, D. M. (2011). Systems Programming. New Delhi: Tata McGraw Hill.
4. Leland Beck., & Manjula, D. (2008). System Software: An Introduction to System Programming (3<sup>rd</sup> ed.). New Delhi: Pearson Education.
5. Grune, D., Van Reeuwijk, K., Bal, H. E., Jacobs, C. J. H., & Langendoen, K.(2012). Modern Compiler Design (2nd ed.). Springer.

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

### Course Objective

- To provide an overview of a new language R used for data science.
- To introduce students to the R programming environment and related eco-system and thus provide them with an in-demand skill-set, in both the research and business environments
- To introduce the extended R ecosystem of libraries and packages
- To demonstrate usage of as standard Programming Language.
- To familiarize students with how various statistics like mean median etc. can be collected for data exploration in R
- To enable students to use R to conduct analytics on large real life datasets.

### Course Outcomes (COs)

1. Install and use R for simple programming tasks.
2. Extend the functionality of R by using add-on packages
3. Extract data from files and other sources and perform various data manipulation tasks on them.
4. Code statistical functions in R.
5. Use R Graphics and Tables to visualize results of various statistical operations on data.
6. Apply the knowledge of R gained to data Analytics for real life applications.

### UNIT- I

**Data Scientist’s Tool Box:** 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.

### UNIT-II

**R Programming Basics:** Overview of R, R data types and objects, reading and writing data, Control structures, functions, scoping rules, dates and times, Loop functions, debugging tools, Simulation, code profiling

### UNIT-III

**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 —tidy.

### UNIT-IV

**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.



## UNIT-V

**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.

### Suggested Readings

1. Rachel Schutt., & Cathy O'Neil.(2013). Doing Data Science: Straight Talk from the Frontline. Schroff/O'Reilly.
2. Foster Provost., & Tom Fawcett.( 2013). Data Science for Business What You Need to Know About Data Mining and Data-Analytic Thinking. O'Reilly.
3. John, W. Foreman. (2013). Data Smart: Using data Science to Transform Information into Insight. John Wiley & Sons.
4. Ian Ayres. (2007). Super Crunchers: Why Thinking-by-Numbers Is the New Way to Be Smart (1st ed.). Bantam.
5. Eric Seigel. (2013). Predictive Analytics: The Power to Predict who Will Click, Buy Lie, or Die (1st ed.). Wiley.
6. Matthew, A. Russel. (2013). Mining the Social Web: Data mining Facebook, Twitter, LinkedIn, Goole+, GitHub, and More (2nd ed.). O'Reilly Media.

---

Instruction Hours / week: L: 3 T: 0 P: 0    **Marks:** Int : 40 Ext : 60    Total: 100

**Course Objectives**

- To gain knowledge about using ORACLE software for developing databases and using them.
- To know the SQL \*plus interface and its commands.
- To create, alter tables, views, indexes ,synonyms and constraints using DDL statements.
- To Query the database using DML statements and write complex queries for information retrieval.
- To gain knowledge about transaction control in Oracle
- To write procedures and functions using PL/SQL.

**Course Outcomes(COs)**

1. Understand the ORACLE software for developing databases and using them.
2. Using the SQL \*plus interface and its commands.
3. creating, altering tables, views, indexes ,synonyms and constraints using DDL statements.
4. Querying the database using DML statements and write complex queries for information retrieval.
5. Apply transaction controls statements in Oracle
6. Write procedures and functions using PL/SQL.

**UNIT-I**

**Introduction to Oracle as RDBMS SQL Vs. SQL \* Plus:** SQL Commands and Data types, Operators and Expressions, Introduction to SQL \* Plus.

**UNIT-II**

**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

**UNIT-III**

**Other Database Objects - View, Synonyms, Index**

**UNIT-IV**

**Transaction Control Statements - Commit, Rollback, Savepoint**

## **UNIT-V**

**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 , Using Cursor (Implicit, Explicit)

### **Suggested Readings**

1. Ivan Bayross. (2010). SQL, PL/SQL the Programming Language of Oracle. New Delhi: BPB Publications.
2. Steven Feuerstein., & Bill Pribyl. (2014). Oracle PL/SQL Programming (6th ed.). O'Reilly Media.
3. Rajeeb, C. Chatterjee. (2012). Learning Oracle SQL and PL/SQL: A simplified Guide. New Delhi: PHI.
4. Ron Hardman.,& Michael Mclaughlin. (2005). Expert Oracle PL/SQL. Oracle Press.
5. Michael Mclaughlin. (2008). Oracle Database 11g PL/SQL Programming. Oracle Press.
6. John Watson.,& Roopesh Ramklass. (2008). OCA Oracle Database11g SQL Fundamentals I Exam Guide. Oracle Press.

---

Instruction Hours / week: L: 3 T: 0 P: 0    **Marks:** Int : 40 Ext : 60    Total: 100

### Course Objectives

- To Learn Syntax and Semantics of Python
- To create Functions in Python.
- To Understand the basic logic statements in Python
- To Handle Strings in Python.
- To Understand Lists, Dictionaries in Python.
- To Build GUI applications

### Course Outcomes (COs)

Upon completion of this the course students will be able to:

1. Develop algorithmic solutions to simple computational problems
2. Structure simple Python programs for solving problems.
3. Learn to use logical constructs in Python
4. Decompose a Python program into functions.
5. Represent compound data using Python lists, tuples, dictionaries.
6. Read and write data from/to files in Python Programs.

### UNIT-I

**Planning the Computer Program:** Concept of problem solving-Problem definition-Program design-Debugging-Types of errors in programming-Documentation.

### UNIT-II

**Techniques of Problem Solving:** Flowcharting-decision table-algorithms-Structured programming concepts-Programming methodologies: top-down and bottom-up Programming.

### UNIT-III

**Overview of Programming:** Structure of a Python Program-Elements of Python.

### UNIT-IV

**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).

### UNIT-V

**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.

## **Suggested Readings**

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

**Course Objectives**

- To provide an overview of Information Security and Assurance.
- To provide an exposure to the spectrum of security activities methods methodologies and procedures with emphasis on practical aspects of Information Security.
- To explain concepts related to applied cryptography including the techniques for crypto-analysis symmetric and asymmetric cryptography.
- To understand the concepts of digital signature, message authentication code, hash functions and modes of encryption operations.
- To explain common vulnerabilities in computer programs including buffer overflow Vulnerabilities time-of-check to time-of-use flaws incomplete mediation.
- To understand the concepts of cryptographic utilities and authentication mechanisms to design secure applications

**Course Outcomes(COs)**

A student who successfully completes this course should at a minimum be able to:

1. Explore the working principles and utilities of various cryptographic algorithms including secret key cryptography, hashes and message digests, and public key algorithms
2. State the basic concepts in information security
3. Explain concepts related to applied cryptography including the techniques for crypto-analysis symmetric and asymmetric cryptography.
4. Use digital signature, message authentication code, hash functions and modes of encryption operations.
5. Explain common vulnerabilities in computer programs including buffer overflow Vulnerabilities time-of-check to time-of-use flaws incomplete mediation.
6. Understand the concepts of cryptographic utilities and authentication mechanisms to design secure applications

**List of programs**

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.

---

**Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60****Total: 100****Course Objectives**

- To describe the fundamental concepts of networks
- To demonstrate mastery of main protocols comprising the Internet.
- To develop skills in network programming techniques.
- To implement network services that communicate through the Internet.
- To apply the client - server model in networking applications.
- To practice networking commands available through the operating systems

**Course Outcomes(COs)**

Having successfully completed this course, the student will be able to:

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

**List of programs**

1. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.
2. Simulate and implement stop and wait protocol for noisy channel.
3. Simulate and implement go back n 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.



---

**Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60****Total: 100****Course Objectives**

- To apply the fundamentals of assembly level programming of microprocessors.
- To build a program on a microprocessor using arithmetic & logical instruction set of 8086.
- To develop the assembly level programming using 8086 loop instruction set.
- To write programs based on string and procedure for 8086 microprocessor.
- To analyze abstract problems and apply a combination of hardware and software to address the problem
- To make use of standard test and measurement equipment to evaluate digital interfaces

**Course Outcomes(COs)**

1. Apply the fundamentals of assembly level programming of microprocessors.
2. Build a program on a microprocessor using arithmetic & logical instruction set of 8086.
3. Develop the assembly level programming using 8086 loop instruction set.
4. Write programs based on string and procedure for 8086 microprocessor.
5. Analyze abstract problems and apply a combination of hardware and software to address the problem
6. Make use of standard test and measurement equipment to evaluate digital interfaces

**Assembly Language Programming**

1. Addition of 8/16 bit and array of Data
2. Subtraction of 8/16 bit and array of Data
3. Multiplication of 8-bit Number
4. Division of 8-bit Number Write a program for 32-bit binary division and multiplication
5. Fill and Transfer an Array of Data
6. Ascending and Descending of an array
7. Data Transfer using Parallel Ports
8. Stepper Motor Interface
9. Traffic Light Interface
10. A/D and D/A Converter

---

**Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60****Total: 100****Course Objectives**

- To understand the fundamental concepts of a digital image processing system.
- To understand concepts of image enhancement techniques.
- To apply various Image Transforms.
- To analyze various Compression techniques and Morphological concepts
- To know various segmentation techniques, and object descriptors.
- To apply color models and various applications of image processing

**Course Outcomes (COs)**

1. Remember the fundamental concepts of image processing.
2. Explain different Image enhancement techniques
3. Understand and review image transforms
4. Analyze the basic algorithms used for image processing & image compression with morphological image processing.
5. Contrast Image Segmentation and Representation
6. Design & Synthesize Color image processing and its real world applications

**List of Programs**

1. Write program to read and display digital image using MATLAB or SCILAB
  - 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 R G & B planes
  - h. Create color image using R, G and B three separate planes
  - i. Flow control and LOOP in SCILAB
  - j. Write given 2-D data in image file
2. To write and execute image processing programs using point processing method
  - a. Obtain Negative image      b. Obtain Flip image
  - b. Thresholding                  d. Contrast stretching
3. To write and execute programs for image arithmetic operations
  - 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
  - a. AND operation between two images

- b. OR operation between two images
  - c. Calculate intersection of two images
  - d. Water Marking using EX-OR operation
  - e. NOT operation (Negative image)
5. To write a program for histogram calculation and equalization using
    - 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
    - a. Translation
    - b. Scaling
    - c. Rotation
    - d. Shrinking
    - e. Zooming
  7. To understand various image noise models and to write programs for
    - 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
    - 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
    - a. Apply FFT on given image
    - b. Perform low pass and high pass filtering in frequency domain
    - c. Apply IFFT to reconstruct image
  10. Write a program in C and MATLAB/SCILAB for edge detection using different edge detection mask
  11. Write and execute program for image morphological operations erosion and dilation.
  12. To write and execute program for wavelet transform on given image and perform inverse wavelet transform to reconstruct image.

**Course Objectives**

- To introduce students to the basic concepts and techniques of Machine Learning.
- To develop skills of using recent machine learning software for solving practical problems.
- To gain experience of doing independent study and research.
- To recognize the characteristics of machine learning that make it useful to real-world problems.
- To characterize machine learning algorithms as supervised, semi-supervised, and unsupervised.
- To effectively use machine learning toolboxes.

**Course Outcomes(COs)**

On successful completion of the course the student should be

1. Remember the basic concepts and techniques of Machine Learning.
2. Develop skills of using recent machine learning software for solving practical problems.
3. of doing independent study and research.
4. To recognize the characteristics of machine learning that make it useful to real-world problems.
5. To characterize machine learning algorithms as supervised, semi-supervised, and unsupervised.
6. To effectively use machine learning toolboxes.

**List of programs**

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 variable.
4. Create/Define single dimension / multi-dimension 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 command 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 type 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 type 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 comprising of 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 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.
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.
15. Use some function for regularization of dataset based on problem 14.
16. Use some function for neural networks, like Stochastic Gradient Descent or back propagation - algorithm to predict the value of a variable based on the dataset of problem

---

**Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60 Total: 100**

**Course Objective**

- To provide an overview of a new language R used for data science.
- To introduce students to the R programming environment and related eco-system and thus provide them with an in-demand skill-set, in both the research and business environments
- To introduce the extended R ecosystem of libraries and packages
- To demonstrate usage of as standard Programming Language.
- To familiarize students with how various statistics like mean median etc. can be collected for data exploration in R
- To enable students to use R to conduct analytics on large real life datasets.

**Course Outcomes (COs)**

1. Install and use R for simple programming tasks.
2. Extend the functionality of R by using add-on packages
3. Extract data from files and other sources and perform various data manipulation tasks on them.
4. Code statistical functions in R.
5. Use R Graphics and Tables to visualize results of various statistical operations on data.
6. Apply the knowledge of R gained to data Analytics for real life applications.

**List of Programs**

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.  
There ages were: 20 20 20 20 20 21 21 21  
22 22 22 22 23 23 23  
i. Find the median age of all  
students under 22 years ii. Find  
the median age of all students

iii. Find the mean age of all students  
iv. Find the modal age for all students

v. 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.

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

12. Obtain probability distribution of  $X$ , where  $X$  is number of spots showing when a six-sided symmetric die (i.e. all six faces of the die are equally likely) is rolled. Simulate random samples of sizes 40, 70 and 100 respectively and verify the frequency interpretation of probability.
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.
14. Use Git / Github software to create Github account. Also, create a repo using Github.

**Course Objectives**

- To gain knowledge about using ORACLE software for developing databases and using them.
- To know the SQL \*plus interface and its commands.
- To create, alter tables, views, indexes ,synonyms and constraints using DDL statements.
- To Query the database using DML statements and write complex queries for information retrieval.
- To gain knowledge about transaction control in Oracle
- To write procedures and functions using PL/SQL.

**Course Outcomes(COs)**

1. Understand the ORACLE software for developing databases and using them.
2. Using the SQL \*plus interface and its commands.
3. creating, altering tables, views, indexes ,synonyms and constraints using DDL statements.
4. Querying the database using DML statements and write complex queries for information retrieval.
5. Apply transaction controls statements in Oracle
6. Write procedures and functions using PL/SQL.

**[SQL COMMANDS]**

- 1) SQL\* formatting commands
- 2) To create a table, alter and drop table.
- 3) To perform select, update, insert and delete operation 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.

**[SQL FUNCTION]**

- 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.

**[PL/SQL]**

- 11) To understand working with PL/SQL
- 12) To implement Cursor on a table.
- 13) To implement trigger on a table



---

**Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60****Total: 100****Course Objectives**

- To Learn Syntax and Semantics of Python
- To create Functions in Python.
- To Understand the basic logic statements in Python
- To Handle Strings in Python.
- To Understand Lists, Dictionaries in Python.
- To Build GUI applications

**Course Outcomes (COs)**

Upon completion of this the course students will be able to:

1. Develop algorithmic solutions to simple computational problems
2. Structure simple Python programs for solving problems.
3. Learn to use logical constructs in Python
4. Decompose a Python program into functions.
5. Represent compound data using Python lists, tuples, dictionaries.
6. Read and write data from/to files in Python Programs.

**List of Programs**

1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon user's 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 :  
Grade A: Percentage  $\geq 80$   
Grade B: Percentage  $\geq 70$  and  $< 80$   
Grade C: Percentage  $\geq 60$  and  $< 70$   
Grade D: Percentage  $\geq 40$  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 user.
4. WAP to display the first n terms of Fibonacci series.
5. WAP to find factorial of the given number.
6. WAP to find sum of the following series for n terms:  $1 - 2/2! + 3/3! - \dots - n/n!$
7. WAP to calculate the sum and product of two compatible matrices.

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

### Course Objectives

- To Provide a good understanding of the concepts, standards in Cloud computing
- To make the student understand about the cloud service providers and their usage.
- To learn how to secure the data in cloud depending.
- To understand the various service level agreements.
- To understand the cloud using various case studies.
- to portray the recent trends in the field of cloud computing and providing exposures to some open source and commercial clouds.

### Course Outcomes (COs)

Upon completion of this course the students will be able to:

1. Portray the recent trends in the field of cloud computing and providing exposures to some open source and commercial clouds.
2. Know the architecture of the cloud and the usage of clouds.
3. Secure their data from the security issues.
4. Make the students to work based on the various service level agreements.
5. Work with the traditional cloud and Microsoft azure, etc.
6. Provide a good understanding of the concepts, standards and protocols in Cloud computing

### UNIT-I

**Overview of Computing Paradigm:** Recent trends in Computing: Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing.**Introduction to Cloud Computing:** Introduction to Cloud Computing, History of Cloud Computing, Cloud service providers, Benefits and limitations of Cloud Computing.

### UNIT-II

**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.

### UNIT-III

**Case Studies:** Case study of Service model using Google App Engine, Microsoft Azure, Amazon EC2 , Eucalyptus.

#### **UNIT-IV**

**Service Management in Cloud Computing:** Service Level Agreements (SLAs), Billing & Accounting, Comparing Scaling Hardware: Traditional vs. Cloud, Economics of scaling.

#### **UNIT-V**

**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.

#### **Suggested Readings**

1. Barrie Sosinsky. (2010). Cloud Computing Bible. New Delhi: Wiley-India,
2. Rajkumar Buyya., James Broberg., & Andrzej, M. Goscinski Wile. Cloud Computing: Principles and Paradigms.
3. Nikos Antonopoulos., & Lee Gillam. (2012). Cloud Computing: Principles, Systems and Applications. Springer.
4. Ronald, L. Krutz., & Russell Dean Vines. (2010). Cloud Security: A Comprehensive Guide to Secure Cloud Computing. New Delhi: Wiley-India.
5. Gautam Shroff. (2010). Enterprise Cloud Computing Technology Architecture Applications. Adobe Reader ebooks available from eBooks.com.
6. Toby Velte., Anthony Velte., & Robert Elsenpeter.(2010). Cloud Computing, A Practical Approach. McGraw Hills.
7. Dimitris, N. Chorafas. (2010). Cloud Computing Strategies. CRC Press.

#### **WEB SITES**

1. [en.wikipedia.org/wiki/Cloud\\_computing](http://en.wikipedia.org/wiki/Cloud_computing)
2. [www.ibm.com/cloud-computing/in/en/](http://www.ibm.com/cloud-computing/in/en/)
3. [www.oracle.com/CloudComputing](http://www.oracle.com/CloudComputing)
4. [www.microsoft.com/en-us/cloud/default.aspx](http://www.microsoft.com/en-us/cloud/default.aspx)

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

### Course Objectives

- To introduce students the concepts and principles of system programming
- To provide students the knowledge about both theoretical and practical aspects of system programming, teaching them the methods and techniques for designing and implementing system-level programs.
- To train students in developing skills for writing system software with the aid of sophisticated OS services, programming languages and utility tools.
- To understand the basics of system programs like editors, compiler, assembler, linker, loader, interpreter and debugger.
- To describe the various concepts of assemblers and macroprocessors.
- To understand how linker and loader create an executable program from an object module created by assembler and compiler.

### Course Outcomes(COs)

1. Understand different components of system software.
2. This course enables for good understanding of the role of system programming and the scope of duties and tasks of a system programmer.
3. This course enables to learn the concepts and principles of developing system-level software (e.g., compiler, and networking software)
4. Understand the basics of system programs like editors, compiler, assembler, linker, loader, interpreter and debugger.
5. Describe the various concepts of assemblers and macroprocessors.
6. Understand how linker and loader create an executable program from an object module created by assembler and compiler.

### UNIT-I

**Assemblers & Loaders, Linkers:** One pass and two pass assembler design of an assembler, Absolute loader, relocation and linking concepts, relocating loader and Dynamic Linking., overview of compilation, Phases of a compiler.

### UNIT-II

#### Lexical Analysis:

Role of a Lexical analyzer, Specification and recognition of tokens, Symbol table, lexical

### UNIT-III

#### Parsing:

Bottom up parsing- LR parser, yacc. **Intermediate representations:** Three address code generation, syntax directed translation, translation of types, control Statements.

### UNIT-IV

**Storage organization:** Activation records stack allocation.

## **UNIT-V**

### **Code Generation:** Object code generation

#### **Suggested Readings**

1. Santanu Chattopadhyaya. (2011). Systems Programming. New Delhi: PHI.
2. Alfred, V. Aho., Monica, S. Lam., Ravi Sethi.,& Jeffrey, D. Ullman. (2006). Compilers: Principles, Techniques, and Tools (2nd ed.). New Delhi: Prentice Hall.
3. Dhamdhare, D. M. (2011). Systems Programming. New Delhi: Tata McGraw Hill.
4. Leland Beck., & Manjula, D. (2008). System Software: An Introduction to System Programming (3<sup>rd</sup> ed.). New Delhi: Pearson Education.
5. Grune, D., Van Reeuwijk, K., Bal, H. E., Jacobs, C. J. H., & Langendoen, K.(2012). Modern Compiler Design (2nd ed.). Springer.

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

**Course Objectives**

- To introduce students to the basic concepts and techniques of Data Mining.
- To understand data mining fundamentals and characterize the kinds of patterns that can be discovered by association rule mining
- To compare and evaluate different data mining techniques like classification, prediction, etc.
- To cluster the high dimensional data for better organization of the data
- To describe complex data types with respect to spatial and web mining
- To design data warehouse with dimensional modelling and apply OLAP operations.

**Course Outcomes (COs)**

Upon completion of this course students will be able to:

1. Understand the basic concepts and techniques of Data Mining
2. Extract knowledge using data mining techniques and Implement Preprocess the data for mining applications and apply the association rules for mining the data
3. Design and deploy appropriate classification techniques
4. Understand the concept of clustering and its real time applications
5. Explore recent trends in data mining such as web mining, spatial-temporal mining
6. Analyze the basic concepts of data warehouse and OLAP operations

**UNIT-I**

**Overview:** Predictive and descriptive data mining techniques

**UNIT-II**

Supervised and unsupervised learning techniques

**UNIT-III**

Process of knowledge discovery in databases, pre-processing methods

**UNIT-IV**

**Data Mining Techniques:** Association Rule Mining, classification and regression techniques, clustering

**UNIT-V**

Scalability and data management issues in data mining algorithms, measures of interestingness.

**Suggested Readings**

1. Pang-Ning Tan., Michael Steinbach.,& Vipin Kumar. (2005).

- Introduction to Data Mining. New Delhi: Pearson Education.
2. Richard Roiger., & Michael Geatz. (2003). Data Mining: A Tutorial Based Primer. New Delhi: Pearson Education.
  3. Gupta, G.K. (2006). Introduction to Data Mining with Case Studies. New Delhi: PHI.
  4. Soman, K. P., Diwakar Shyam., & Ajay, V. (2006). Insight Into Data Mining: Theory And Practice. New Delhi: PHI.

#### **WEB SITES**

1. Thedacs.Com
2. Dwreview.Com
3. Pcai.Com
4. Eruditionhome.Com

---

**Instruction Hours / week: L: 4 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

**Course Objectives**

- To describe characteristics and functioning of common graphics input/output devices
- To learn the basic principles of 3- dimensional computer graphics
- To Provide an understanding of how to scan convert the basic geometrical primitives, how to transform the shapes to fit them as per the picture definition
- To Provide an understanding of mapping from a world coordinates to device coordinates and projections.
- To extract scene with different clipping methods and its transformation to graphics display device.
- To explore projections and visible surface detection techniques for display of 3D scene on 2D screen

**Course Outcomes(COs)**

1. Understand the basics of computer graphics, different graphics systems and applications of computer graphics.
2. Discuss various algorithms for scan conversion and filling of basic objects and their comparative analysis.
3. Use of geometric transformations on graphics objects and their application in composite form.
4. Extract scene with different clipping methods and its transformation to graphics display device.
5. Explore projections and visible surface detection techniques for display of 3D scene on 2D screen.
6. Render projected objects to naturalize the scene in 2D view and use of illumination models for this..

**UNIT-I**

**Introduction :** Basic elements of Computer graphics, Applications of Computer Graphics.

**UNIT-II**

**Graphics Hardware :** Architecture of Raster and Random scan display devices, input/output devices.

**UNIT-III**

**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.

**UNIT-IV**

**Geometric Modeling :** Representing curves & Surfaces.



## **UNIT V**

**Visible Surface determination :** Hidden surface elimination. **Surface rendering :** Illumination and shading models. Basic color models and Computer Animation.

### **Suggested Readings**

1. Foley, J.D., Van Dam, A., & Feiner Hughes. (1990). Computer Graphics Principles & Practice (2nd ed.). Addison Wesley.
2. Hearn D. Baker. (2008). Computer Graphics. New Delhi: Prentice Hall of India.
3. Rogers, D.F. (1997). Procedural Elements for Computer Graphics. McGraw Hill.
4. Rogers, D.F. Adams. (1989). Mathematical Elements for Computer Graphics(2<sup>nd</sup> ed.). McGraw Hill.

---

**Instruction Hours / week: L: 3 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

### Course Objectives

- To write basic PHP syntax using various operators.
- To write PHP scripts to handle HTML forms.
- To analyze different tasks using PHP functions.
- To understand the regular expressions in PHP.
- To learn array data structure using PHP scripts.
- To work with open source applications that deal with database and website development.

### Course Outcomes (COs)

Upon completion of this course, students will be able to:

1. Write PHP scripts using operators to perform various functions
2. Design PHP scripts to handle HTML forms.
3. Implement different types of PHP functions.
4. Write regular expressions including modifiers, operators, and metacharacters.
5. Create PHP scripts using array.
6. Develop dynamic web pages.

### UNIT-I

#### Introduction to PHP:

- PHP introduction, inventions and versions, important tools and software requirements (like Web Server, Database, Editors etc.)
- PHP with other technologies, scope of PHP
- Basic Syntax, PHP variables and constants
- Types of data in PHP , Expressions, scopes of a variable (local, global)
- PHP Operators: Arithmetic, Assignment, Relational, Logical operators, Bitwis , ternary and MOD operator.
- PHP operator Precedence and associativity

### UNIT-II

#### Handling HTML form with PHP:

- Capturing Form Data
- GET and POST form methods
- Dealing with multi value fields
- Redirecting a form after submission

#### PHP conditional events and Loops:

- PHP IF Else conditional statements ( Nested IF and Else)
- Switch case, while ,For and Do While Loop
- Goto , Break ,Continue and exit

### **UNIT-III**

#### **PHP Functions:**

- Function, Need of Function , declaration and calling of a function
- PHP Function with arguments, Default Arguments in Function
- Function argument with call by value, call by reference
- Scope of Function Global and Local

### **UNIT-IV**

#### **String Manipulation and Regular Expression:**

- Creating and accessing String , Searching & Replacing String
- Formatting, joining and splitting String , String Related Library functions
- Use and advantage of regular expression over inbuilt function
- Use of preg\_match(), preg\_replace(), preg\_split() functions in regular expression

### **UNIT-V**

#### **Array:**

- Anatomy of an Array ,Creating index based and Associative array ,Accessing array
- Looping with Index based array, with associative array using each() and foreach()
- Some useful Library function

#### **Suggested Readings**

1. Steven Holzner. (2007). PHP: The Complete Reference. New Delhi: McGraw Hill Education (India).
2. Timothy Boronczyk., & Martin, E. Psinas. (2008). PHP and MYSQL (Create-Modify-Reuse). New Delhi: Wiley India Private Limited.
3. Robin Nixon. (2014). Learning PHP, MySQL, JavaScript, CSS & HTML5 (3rd ed.). O'reilly.
4. Luke Welling.,& Laura Thompson.(2008). PHP and MySQL Web Development (4th ed.). Addition Paperback, Addison-Wesley Professional.
5. David Sklar., & Adam Trachtenberg. PHP Cookbook: Solutions & Examples for PHP.

---

**Instruction Hours / week: L: 3 T: 0 P: 0    Marks: Int : 40 Ext : 60    Total: 100**

**Course Objective**

- To understand fundamental operating system abstractions such as processes, threads, files, semaphores, IPC abstractions, shared memory regions, etc.
- To understand how the operating system abstractions can be used in the development of application programs, or to build higher level abstractions
- To understand how the operating system abstractions can be implemented
- To understand the principles of concurrency and synchronization, and apply them to write correct concurrent programs/software
- To understand basic resource management techniques like scheduling or time management, space management and principles and how they can be implemented.
- To know about the issues of performance and fairness objectives, avoiding deadlocks, as well as security and protection.

**Course Outcomes(COs)**

1. This course teaches the student the concepts and principles that underlie modern operating systems, and a practice component to relate theoretical principles with operating system implementation.
2. Implement operating system abstractions in the development of application programs
3. Apply the principles of concurrency and synchronization to write correct concurrent programs/software
4. Implement basic resource management techniques like scheduling or time management, space management
5. Learn about processes and processor management, memory management schemes, file system and secondary storage management.
6. Learn about issues of performance and fairness objectives, avoiding deadlocks, as well as security and protection

**UNIT-I**

**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

**UNIT-II**

System processes (an overview), External and internal commands, Creation of partitions in OS, Processes and its creation phases – Fork, Exec, wait

**UNIT-III**

**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)

#### **UNIT-IV**

**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)

#### **UNIT-V**

**System calls, Using system calls** Pipes and Filters, Decision making in Shell Scripts (If else, switch), Loops in shell Functions, Utility programs (cut, paste, join, tr, uniq utilities), Pattern matching utility (grep)

#### **Suggested Readings**

1. Sumitabha, Das.(2006). Unix Concepts And Applications. New Delhi: Tata McGraw-Hill Education.
2. Michael Jang. (2011). RHCSA/ RHCE Red Hat Linux Certification: Exams (Ex200 & Ex300). Certification Press.
3. Nemeth Synder., & Hein.(2010). Linux Administration Handbook (2nd ed.). Pearson Education.
4. Richard Stevens, W. Bill Fenner., & Andrew, M. Rudoff. (2014). Unix Network Programming, The sockets Networking API, Vol. 1, (3rd ed.).

**Course Objectives**

- To Provide a good understanding of the concepts, standards in Cloud computing
- To make the student understand about the cloud service providers and their usage.
- To learn how to secure the data in cloud depending.
- To understand the various service level agreements.
- To understand the cloud using various case studies.
- to portray the recent trends in the field of cloud computing and providing exposures to some open source and commercial clouds.

**Course Outcomes (COs)**

Upon completion of this course the students will be able to:

1. Portray the recent trends in the field of cloud computing and providing exposures to some open source and commercial clouds.
2. Know the architecture of the cloud and the usage of clouds.
3. Secure their data from the security issues.
4. Make the students to work based on the various service level agreements.
5. Work with the traditional cloud and Microsoft azure, etc.
6. Provide a good understanding of the concepts, standards and protocols in Cloud computing

**List of programs**

1. Create virtual machines that access different programs on same platform.
2. Create virtual machines that access different programs on different platforms.
3. Working on tools used in cloud computing online-
  - a) Storage
  - b) Sharing of data
  - c) Manage your calendar, to-do lists,
  - d) A document editing tool
4. Exploring Google cloud
5. Exploring Microsoft cloud
6. Exploring Amazon cloud

---

**Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60****Total: 100****Course Objectives**

- To introduce students the concepts and principles of system programming
- To provide students the knowledge about both theoretical and practical aspects of system programming, teaching them the methods and techniques for designing and implementing system-level programs.
- To train students in developing skills for writing system software with the aid of sophisticated OS services, programming languages and utility tools.
- To understand the basics of system programs like editors, compiler, assembler, linker, loader, interpreter and debugger.
- To describe the various concepts of assemblers and macroprocessors.
- To understand how linker and loader create an executable program from an object module created by assembler and compiler.

**Course Outcomes(COs)**

1. Understand different components of system software.
2. This course enables for good understanding of the role of system programming and the scope of duties and tasks of a system programmer.
3. This course enables to learn the concepts and principles of developing system-level software (e.g., compiler, and networking software)
4. Understand the basics of system programs like editors, compiler, assembler, linker, loader, interpreter and debugger.
5. Describe the various concepts of assemblers and macroprocessors.
6. Understand how linker and loader create an executable program from an object module created by assembler and compiler.

**List of Programs**

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.

---

**Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60 Total: 100**

**Course Objectives**

- To introduce students to the basic concepts and techniques of Data Mining.
- To understand data mining fundamentals and characterize the kinds of patterns that can be discovered by association rule mining
- To compare and evaluate different data mining techniques like classification, prediction, etc.
- To cluster the high dimensional data for better organization of the data
- To describe complex data types with respect to spatial and web mining
- To design data warehouse with dimensional modelling and apply OLAP operations.

**Course Outcomes (COs)**

Upon completion of this course students will be able to:

7. Understand the basic concepts and techniques of Data Mining
8. Extract knowledge using data mining techniques and Implement Preprocess the data for mining applications and apply the association rules for mining the data
9. Design and deploy appropriate classification techniques
10. Understand the concept of clustering and its real time applications
11. Explore recent trends in data mining such as web mining, spatial-temporal mining
12. Analyze the basic concepts of data warehouse and OLAP operations

**Practical exercises based on concepts listed in theory.**



---

**Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60****Total: 100****Course Objectives**

- To describe characteristics and functioning of common graphics input/output devices
- To learn the basic principles of 3- dimensional computer graphics
- To Provide an understanding of how to scan convert the basic geometrical primitives, how to transform the shapes to fit them as per the picture definition
- To Provide an understanding of mapping from a world coordinates to device coordinates and projections.
- To extract scene with different clipping methods and its transformation to graphics display device.
- To explore projections and visible surface detection techniques for display of 3D scene on 2D screen

**Course Outcomes(COs)**

1. Understand the basics of computer graphics, different graphics systems and applications of computer graphics.
2. Discuss various algorithms for scan conversion and filling of basic objects and their comparative analysis.
3. Use of geometric transformations on graphics objects and their application in composite form.
4. Extract scene with different clipping methods and its transformation to graphics display device.
5. Explore projections and visible surface detection techniques for display of 3D scene on 2D screen.
6. Render projected objects to naturalize the scene in 2D view and use of illumination models for this..

**List of Programs**

1. Write a program to implement Bresenham's line drawing algorithm.
2. Write a program to implement mid-point 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 object (use homogenous 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 Hermite/Bezier curve.

---

**Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Int : 40 Ext : 60****Total: 100****Course Objectives**

- To write basic PHP syntax using various operators.
- To write PHP scripts to handle HTML forms.
- To analyze different tasks using PHP functions.
- To understand the regular expressions in PHP.
- To learn array data structure using PHP scripts.
- To work with open source applications that deal with database and website development.

**Course Outcomes (COs)**

Upon completion of this course, students will be able to:

1. Write PHP scripts using operators to perform various functions
2. Design PHP scripts to handle HTML forms.
3. Implement different types of PHP functions.
4. Write regular expressions including modifiers, operators, and metacharacters.
5. Create PHP scripts using array.
6. Develop dynamic web pages.

**List of programs**

1. Create a PHP page using functions for comparing three integers and print the largest number.
2. Write a function to calculate the factorial of a number (non-negative integer). The function aCSUept the number as an argument.
3. WAP to check whether the given number is prime or not.
4. Create a PHP page which aCSUepts string from user. After submission that page displays the reverse of provided string.
5. Write a PHP function that checks if a string is all lower case.
6. Write a PHP script that checks whether a passed string is palindrome or not? ( A palindrome is word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run)
7. WAP to sort an array.
8. Write a PHP script that removes the whitespaces from a string.  
Sample string : 'The quick brown fox' Expected Output : Thequickbrownfox

9. Write a PHP script that finds out the sum of first n odd numbers.
  10. Create a login page having user name and password. On clicking submit, a welcome message should be displayed if the user is already registered (i.e.name is present in the database) otherwise error message should be displayed.
  11. Write a PHP script that checks if a string contains another string.
  12. Create a simple 'birthday countdown' script, the script will count the number of days between current day and birth day.
  13. Create a script to construct the following pattern, using nested for loop.
- ```

*
* *
* * *
* * * *
* * * * *

```
14. Write a simple PHP program to check that emails are valid.
  15. WAP to print first n even numbers.
  16. \$color = array('white', 'green', 'red')

Write a PHP script which will display the colors in the following way : Output :

```

white, green, red,
• green • red
• white

```

17. Using switch case and dropdown list display a —Hello! message depending on the language selected in drop down list.
18. Write a PHP program to print Fibonacci series using recursion.
19. Write a PHP script to replace the first 'the' of the following string with 'That'.

**Sample :** 'the quick brown fox jumps over the lazy dog.'  
**Expected Result :** That quick brown fox jumps over the lazy dog.

---

**Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Int : 40 Ext : 60 Total: 100**

### Course Objective

- To understand fundamental operating system abstractions such as processes, threads, files, semaphores, IPC abstractions, shared memory regions, etc.
- To understand how the operating system abstractions can be used in the development of application programs, or to build higher level abstractions
- To understand how the operating system abstractions can be implemented
- To understand the principles of concurrency and synchronization, and apply them to write correct concurrent programs/software
- To understand basic resource management techniques like scheduling or time management, space management and principles and how they can be implemented.
- To know about the issues of performance and fairness objectives, avoiding deadlocks, as well as security and protection.

### Course Outcomes(COs)

1. This course teaches the student the concepts and principles that underlie modern operating systems, and a practice component to relate theoretical principles with operating system implementation.
2. Implement operating system abstractions in the development of application programs
3. Apply the principles of concurrency and synchronization to write correct concurrent programs/software
4. Implement basic resource management techniques like scheduling or time management, space management
5. Learn about processes and processor management, memory management schemes, file system and secondary storage management.
6. Learn about issues of performance and fairness objectives, avoiding deadlocks, as well as security and protection

### List of programs

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 modify —call command to display calendars of the specified months.
3. Write a shell script to modify —call command 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 display message —Entered login name is invalid.
5. Write a shell script to display date in the mm/dd/yy format.
6. Write a shell script to display on the screen sorted output of —who 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 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 have all the permissions or not

**Course Objectives**

- To provide an overview of an exciting growing field of Big Data analytics.
- To discuss the challenges traditional data mining algorithms face when analyzing Big Data.
- To introduce the tools required to manage and analyze big data like Hadoop, NoSql MapReduce.
- To teach the fundamental techniques and principles in achieving big data analytics with scalability and streaming capability.
- To introduce to the students several types of big data like social media, web graphs and data streams.
- To enable students to have skills that will help them to solve complex real-world problems in for decision support

**Course Outcomes(COs)**

1. Explain the motivation for big data systems and identify the main sources of Big Data in the real world.
2. Demonstrate an ability to use frameworks like Hadoop, NOSQL to efficiently store retrieve and process Big Data for Analytics.
3. Implement several Data Intensive tasks using the Map Reduce Paradigm
4. Apply several newer algorithms for Clustering Classifying and finding associations in Big Data
5. Design algorithms to analyze Big dat a like streams, Web Graphs and Social Media data.
6. Design and implement successful Recommendation engines for enterprises.

**UNIT-I**

Fundamentals of Big Data - The Evolution of Data Management Understanding the Waves of Managing Data- Defining Big Data - Big Data Management Architecture- The Big Data Journey -Big Data Types-Defining Structured Data-Defining Unstructured Data-Putting Big Data Together.

**UNIT-II**

Big Data Stack- Basics of Virtualization - The importance of virtualization to big data - Server virtualization - Application virtualization - Network virtualization -Processor and memory virtualization - Data and storage virtualization-Abstraction and Virtualization-Implementing Virtualization to Work with Big Data.

**UNIT-III**

Hadoop - Hadoop Distributed File System - Hadoop MapReduce- The Hadoop foundation and Ecosystem.

#### **UNIT-IV**

Big Data Analytics-Text Analytics and Big Data-Customized Approaches for Analysis of Big Data

#### **UNIT-V**

Integrating Data Sources-Real-Time Data Streams and Complex Event Processing-Operationalizing Big Data.

#### **Suggested Readings :**

1. Judith Hurwitz, Alan Nugent, Fern Halper, Marcia Kaufman (2013). Big Data For Dummies. New Delhi: Wiley India.
2. Paul Zikopoulos, Dirk deRoos, Krishnan Parasuraman, Thomas Deutsch, James Giles, David Corrigan (2012). Harness the Power of Big Data The IBM Big Data Platform. New Delhi: Tata McGraw Hill Publications.
3. Michael Minelli, Michele Chambers, Ambiga Dhiraj (2013). Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses. New Delhi: Wiley Publications.
4. Zikopoulos, Paul, Chris Eaton (2011). Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. New Delhi: Tata McGraw Hill Publications.

#### **WEB SITES**

1. [www.oracle.com/BigData](http://www.oracle.com/BigData)
2. [www.planet-data.eu/sites/default/files/Big\\_Data\\_Tutorial\\_part4.pdf](http://www.planet-data.eu/sites/default/files/Big_Data_Tutorial_part4.pdf)
3. [www.ibm.com/developerworks/data](http://www.ibm.com/developerworks/data)
4. [www.solacesystems.com](http://www.solacesystems.com)
5. [en.wikipedia.org/wiki/Big\\_data](http://en.wikipedia.org/wiki/Big_data)
6. [www.sap.com/solution/big-data.html](http://www.sap.com/solution/big-data.html)

---

**Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : 40 Ext : 60 Total: 100**

**Course Objectives**

- To provide an overview of an exciting growing field of Big Data analytics.
- To discuss the challenges traditional data mining algorithms face when analyzing Big Data.
- To introduce the tools required to manage and analyze big data like Hadoop, NoSql MapReduce.
- To teach the fundamental techniques and principles in achieving big data analytics with scalability and streaming capability.
- To introduce to the students several types of big data like social media, web graphs and data streams.
- To enable students to have skills that will help them to solve complex real-world problems in for decision support

**Course Outcomes(COs)**

1. Explain the motivation for big data systems and identify the main sources of Big Data in the real world.
2. Demonstrate an ability to use frameworks like Hadoop, NOSQL to efficiently store retrieve and process Big Data for Analytics.
3. Implement several Data Intensive tasks using the Map Reduce Paradigm
4. Apply several newer algorithms for Clustering Classifying and finding associations in Big Data
5. Design algorithms to analyze Big dat a like streams, Web Graphs and Social Media data.
6. Design and implement successful Recommendation engines for enterprises.

**List of Programs**

1. Implement a quicksort using scala.
2. Implement a auction service using scala.
3. Write a scala function to perform any 10 arithmetic operations.
4. Write a program to find the factorial of a given number using recursion.
5. Write a program for string manipulations.
6. Write a program for alphabetic order arrangement of a set of names.
7. Write a program for student records using scala list.
8. Implement any 5 map methods for maintaining customer details.
9. Implement employee records using Files
10. Write a program to copy the files using command line arguments.