B.Sc. COMPUTER SCIENCE

CHOICE BASED CREDIT SYSTEM (CBCS)

Curriculum and Syllabus
Regular (2019 – 2020)

Enable | Enlighten | Enrich
KARPAGAM
ACADEMY OF HIGHER EDUCATION
(Deemed to be University)
(Established Under Section 3 of UGC Act, 1956)

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: 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		

DEPARTMENT OF COMPUTER SCIENCE, COMPUTER APPLICATIONS & INFORMATION TECHNOLOGY
FACULTY OF ARTS, SCIENCE AND HUMANITIES
UG PROGRAM (CBCS) – B.Sc. Computer Science
(2019–2020 Batch and onwards)

Course code	Name of the course	Objectives and out comes		Instruction hours / week			Credit(s)	Maximum Marks		
		PEOs	POs	L	T	P		CIA	ESE	Total
								40	60	100
SEMESTER - I										
19LSU101	Language-I	IV	d,e	04	-	-	4	40	60	100
19CSU101	Programming Fundamentals using C / C++	I	a,b,c	05	-	-	5	40	60	100
19CSU102	Computer System Architecture	I	b,c,g	04	-	-	4	40	60	100
19CSU103	Computer Fundamentals	III	h,j	04	-	-	4	40	60	100
19CSU111	Programming Fundamentals using C / C++ - Practical	I	a,b,c,g	-	-	04	2	40	60	100
19CSU112	Computer System Architecture – Practical	I	a,c,g	-	-	03	2	40	60	100
19CSU113	Computer Fundamentals – Practical	III	b,h,j	-	-	03	2	40	60	100
19AEC101	Environmental Studies	IV	d,e	03	-	-	3	40	60	100
Semester Total				**20**	**-**	**10**	**26**	**320**	**480**	**800**
SEMESTER – II										
19LSU201	Language – II	IV	d,e	04	-	-	4	40	60	100
19ENU201	English – I	II	d,f	04	-	-	4	40	60	100
19CSU201	Programming in JAVA	I	c,h,i	04	-	-	4	40	60	100
19CSU202	Discrete Structures	III	a,b	04	-	-	4	40	60	100
19CSU203	Computer Networks and Internet Technologies	IV	e,i	04	-	-	4	40	60	100
19CSU211	Programming in JAVA – Practical	I	a,c,h,i	-	-	04	2	40	60	100
19CSU212	Discrete Structures – Practical	III	a,b,j	-	-	03	2	40	60	100
19CSU213	Computer Networks and Internet Technologies - Practical	IV	c,e	-	-	03	2	40	60	100
Semester Total				**20**	**-**	**10**	**26**	**320**	**480**	**800**
SEMESTER - III										
19CSU301	Data Structures	I	a,b,g, h	04	-	-	4	40	60	100
19CSU302	Operating Systems	III	a,b,h, k	04	-	-	4	40	60	100
19CSU303	Computer Networks	III	a,b,j,k	04	-	-	4	40	60	100
19CSU304A	Android Programming	I	a,b,c, m	03	-	-	3	40	60	100

19CSU304B	Programming in Visual Basic / Gambas	IV	c,d,e,i					40	60	100
19CSU311	Data Structures – Practical	I	a,b,g,h	-	-	04	2	40	60	100
19CSU312	Operating Systems – Practical	III	a,b,h,k	-	-	04	2	40	60	100
19CSU313	Computer Networks – Practical	III	a,b,j,k	-	-	04	2	40	60	100
19CSU314A	Android Programming – Practical	I	a,b,c,m	-	-	03	1	40	60	100
19CSU314B	Programming in Visual Basic / Gambas - Practical	IV	c,d,e,i	-	-	03		40	60	100
Semester Total				**15**	**-**	**15**	**22**	**320**	**480**	**800**
SEMESTER – IV										
19CSU401	Design and Analysis of Algorithms	I	a,b,c,m	04	-	-	4	40	60	100
19CSU402	Software Engineering	IV	c,d,e,l	04	-	-	4	40	60	100
19CSU403	Database Management Systems	I	a,b,g,h	04	-	-	4	40	60	100
19CSU404A	HTML Programming	III	a,b,h,j,k	03	-	-	3	40	60	100
19CSU404B	XML Programming	III	a,b,h,j,k							
19CSU411	Design and Analysis of Algorithms - Practical	I	a,b,c,m	-	-	04	2	40	60	100
19CSU412	Software Engineering – Practical	IV	c,d,e,l	-	-	04	2	40	60	100
19CSU413	Database Management Systems – Practical	I	a,b,g,h	-	-	04	2	40	60	100
19CSU414A	HTML Programming – Practical	III	a,b,h,j,k			03	1	40	60	100
19CSU414B	XML Programming – Practical	III	a,b,h,j,k							
Semester Total				**15**	**-**	**15**	**22**	**320**	**480**	**800**
SEMESTER –V										
19CSU501A	Cloud Computing	I	b,e,m	04	-	-	4	40	60	100
19CSU501B	Software Testing	I	c,g		-	-				
19CSU502A	Internet Technologies	III	a,b,h,j	04	-	-	4	40	60	100
19CSU502B	Information Security and Cyber Law	I	a,b,h							
19CSU503A	Data Mining	III	A,b,h,k	04	-	-	4	40	60	100
19CSU503B	R Programming	II	d,e,f		-	-				
19CSU504A	Oracle (SQL/PL-SQL)	IV	c,e,i,l	03	-	-	3	40	60	100
19CSU504B	Programming in Python	III	b,h,j,k							
19CSU511A	Cloud Computing – Practical	I	b,e,m	-	-	04	2	40	60	100
19CSU511B	Software Testing - Practical	I	c,g	-	-					

19CSU512A	Internet Technologies - Practical	III	a,b,h,j	-	-	04	2	40	60	100
19CSU512B	Information Security and Cyber Law - Practical	I	a,b,h							
19CSU513A	Data Mining – Practical	III	a,b,h, k	-	-	04	2	40	60	100
19CSU513B	R Programming – Practical	II	d,e,f	-	-					
19CSU514A	Oracle (SQL/PL-SQL) – Practical	IV	c,e,i,l	-	-	03	1	40	60	100
19CSU514B	Programming in Python – Practical	III	b,h,j,k							
	Semester Total			**15**	**-**	**15**	**22**	**320**	**480**	**800**
SEMESTER –VI										
19CSU601A	PHP Programming	V	e,j,l	04	-	-	4	40	60	100
19CSU601B	Unix / Linux Programming	I	a,b,h,i							
19CSU602A	Web and E-Commerce Technologies	I	a,d,g, m	04	-	-	4	40	60	100
19CSU602B	Computer Graphics	I	a,c,g, m							
19CSU603A	Artificial Intelligence	III	a,b,h,j ,k	03	-	-	3	40	60	100
19CSU603B	System Programming	IV	c,d,e							
19CSU611A	PHP Programming –Practical	V	e,j,l	-	-	04	2	40	60	100
19CSU611B	Unix / Linux Programming – Practical	I	a,b,h,i							
19CSU612A	Web and E-Commerce Technologies - Practical	I	a,d,g, m	-	-	04	2	40	60	100
19CSU612B	Computer Graphics – Practical	I	a,c,g, m	-	-					
19CSU613A	Artificial Intelligence – Practical	III	a,b,h,j ,k	-	-	03	1	40	60	100
19CSU613B	System Programming – Practical	IV	c,d,e							
19CSU691	Project	II	d,e,f,n	08	-	-	6	40	60	100
	ECA / NCC / NSS / Sports / General interest etc	Good								
	Semester Total			**19**	**-**	**11**	**22**	**280**	**420**	**700**
	Grand Total			**104**	**-**	**76**	**140**	**1880**	**2820**	**4700**

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

Ability Enhancement Courses (AEC)		
Semester	Course Code	Name of the Course
I	19LSU101	Language –I
	19AEC101	Environmental Studies
II	19LSU201	Language –II
	19ENU201	English

Generic Elective Courses (GE) /Allied Courses		
Semester	Course Code	Name of the Course
I	19CSU102	Computer System Architecture
	19CSU112	Computer System Architecture - Practical
II	19CSU202	Discrete Structures
	19CSU212	Discrete Structures - Practical

Core Courses (CC)		
Semester	Course Code	Name of the Course
I	19CSU101	Programming Fundamentals using C / C++
	19CSU103	Computer Fundamentals
	19CSU111	Programming Fundamentals using C / C++ -Practical
	19CSU113	Computer Fundamentals - Practical
II	19CSU201	Programming in JAVA
	19CSU203	Computer Networks and Internet Technologies
	19CSU211	Programming in JAVA - Practical
	19CSU213	Computer Networks and Internet Technologies - Practical
III	19CSU301	Data Structures
	19CSU302	Operating Systems
	19CSU303	Computer Networks
	19CSU311	Data Structures – Practical
	19CSU312	Operating Systems – Practical
	19CSU313	Computer Networks– Practical
IV	19CSU401	Design and Analysis of Algorithms
	19CSU402	Software Engineering
	19CSU403	Database Management Systems
	19CSU411	Design and Analysis of Algorithms - Practical
	19CSU412	Software Engineering – Practical
	19CSU413	Database Management Systems – Practical
V	19CSU502A	Internet Technologies
	19CSU502B	Information Security and Cyber Laws
	19CSU512A	Internet Technologies– Practical
	19CSU512B	Information Security and Cyber Laws – Practical
VI	19CSU603A	Artificial Intelligence
	19CSU603B	System Programming
	19CSU613A	Artificial Intelligence – Practical
	19CSU613B	System Programming – Practical
	19CSU691	Project

Skill Enhancement Courses(SEC)		
Semester	Course Code	Name of the Course
III	19CSU304A	Android Programming
	19CSU304B	Programming in Visual Basic/Gambas
	19CSU314A	Android Programming – Practical
	19CSU314B	Programming in Visual Basic/Gambas– Practical
IV	19CSU404A	HTML Programming
	19CSU404B	XML Programming
	19CSU414A	HTML Programming – Practical
	19CSU414B	XML Programming – Practical
V	19CSU501A	Cloud Computing
	19CSU501B	Software Testing
	19CSU511A	Cloud Computing - Practical
	19CSU511B	Software Testing – Practical
VI	19CSU601A	PHP Programming
	19CSU601B	Unix / Linux Programming
	19CSU611A	PHP Programming –Practical
	19CSU611B	Unix / Linux Programming – Practical

Discipline Specific Elective Courses (DSE)		
Semester	Course Code	Name of the Course
V	19CSU503A	Data Mining
	19CSU503B	R Programming
	19CSU504A	Oracle (SQL/PL-SQL)
	19CSU504B	Programming in Python
V	19CSU513A	Data Mining – Practical
	19CSU513B	R Programming –Practical
	19CSU514A	Oracle (SQL/PL-SQL) – Practical
	19CSU514B	Programming in Python – Practical
VI	19CSU602A	Web and E-Commerce Technologies
	19CSU602B	Computer Graphics
	19CSU612A	Web and E-Commerce Technologies – Practical
	19CSU612B	Computer Graphics – Practical

கற்பகம்உயர்கல்விகலைக்கழகம்

தமிழ்த்துறை

பகுதி - I தமிழ்ப்பாடத்திட்டம் (2019 - 2020)

முதல்பருவம்

(இளநிலை அறிவியல் பட்டவகுப்புகளுக்குரியது)

(For I-UG Science Degree Classes) 19LSU101

பாடத்திட்டப்பொதுநோக்கம்

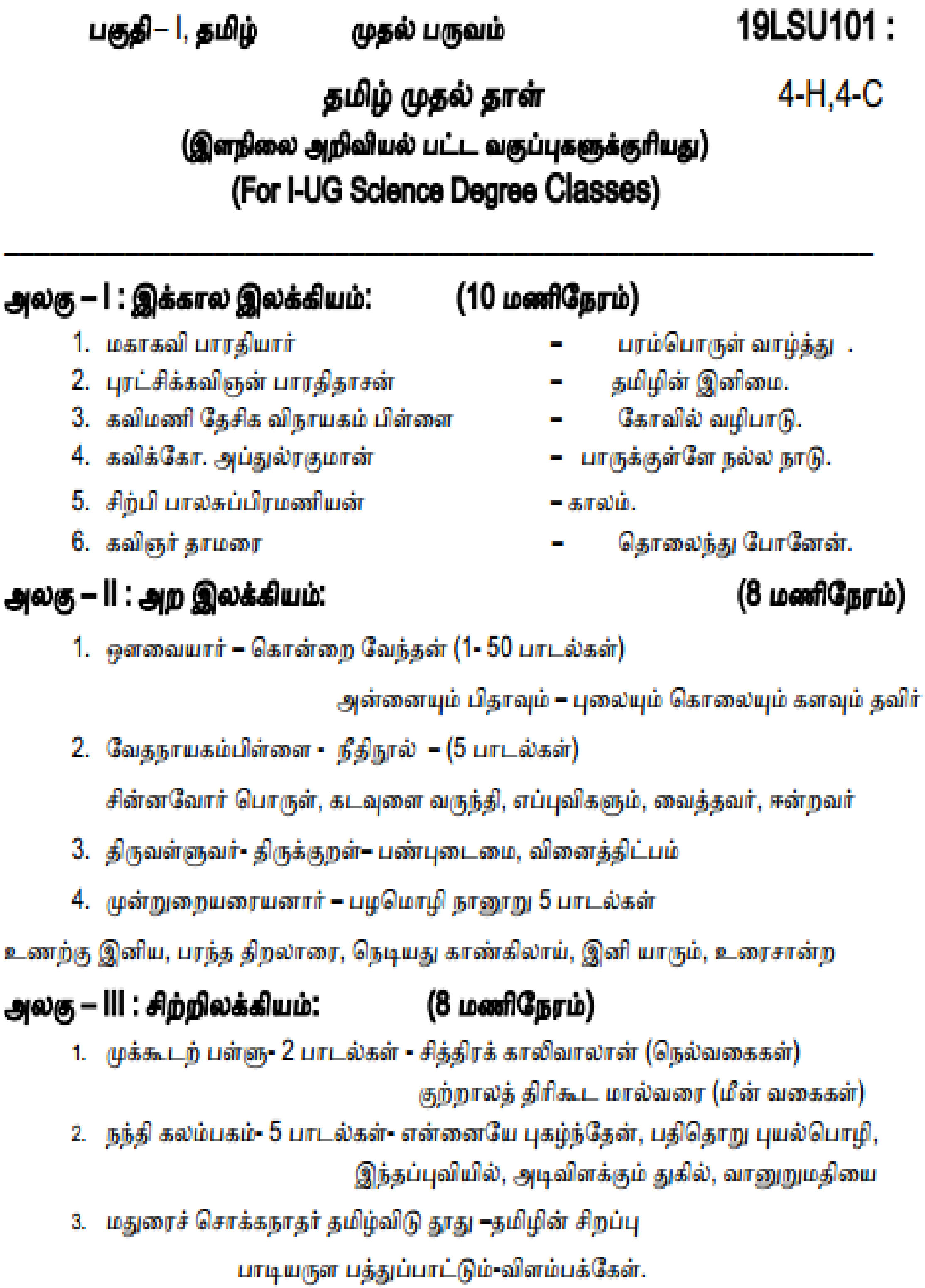
- கற்றல் வழி சிந்தனைத் திறனையும், கருத்து வெளிப்பாட்டுத் திறனையும், மேம்படுத்துதல்.
- ஆய்வுநோக்கைமேம்படுத்துதல்.
- இலக்கியங்கள் உணர்த்தும் வாழ்வின் நுட்பமான பகுதிகளை உணர்த்துதல்.
- மனித மனத்தினைப் பக்குவப்படுத்துதலில் இலக்கியம் தரும் பங்கினை உணர்த்துதல்.
- வளர்ந்து வரும் சமூகத்தில் அறஉணர்வு, பண்பாடு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

பாடத்திட்டப்பயன்விளைவு

- இந்தியகுடியுரிமைப்பணிமுதலானபோட்டித்தேர்வுகளில், விருப்பப்பாடமாகஇடம்பெறுகின்ற, ‘ தமிழ்இலக்கியவரலாறு’ குறித்தமுழுமையானஅறிமுகம்பெற்றிருத்தல்.
- கல்வெட்டியல், ஓலைச்சுவடியியல்மற்றும்தொல்லியல்சார்ந்தஆவணத்தேடலுக்குரியஆய்வுமனப்பான்மையுடன், இலக்கியங்களைஅணுகுதல்.
- தமிழின்வளர்ச்சித்துறையாகிய, ‘அறிவியல்தமிழ்’ ; ‘இணையதமிழ்’ குறித்தபன்நோக்குஅணுகுமுறையிலானஆய்வுச்சிந்தனைமேம்பாடு.
- வேலைவாய்ப்புக்குரியசுயதிறன்மேம்பாட்டுடன், படைப்பாக்கத்திறன்மேம்பாடும்பெற்றிருத்தல் .
- சமுதாயமற்றும்வாழ்வியல்மதிப்புகளைப்பேணுவதற்குக்கருவியாகஇலக்கியங்களைநாடுகின்றமனப் பான்மைவளர்ச்சி.
- மொழிபெயப்புத்துறைசார்ந்தவேலைவாய்புத்திறன்பெற்றிருத்தல்

தாள்கள்வரிசையும்தேர்வுச்செயல்திட்டமும்பகுதி-I தமிழ்

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



பகுதி– I, தமிழ் முதல் பருவம் 19LSU101 :

தமிழ் முதல் தாள் 4-H,4-C

(இளநிலை அறிவியல் பட்ட வகுப்புகளுக்குரியது)

(For I-UG Science Degree Classes)

அலகு – I : இக்கால இலக்கியம்: (10 மணிநேரம்)

1. மகாகவி பாரதியார் - பரம்பொருள் வாழ்த்து .
2. புரட்சிக்கவிஞன் பாரதிதாசன் - தமிழின் இனிமை.
3. கவிமணி தேசிக விநாயகம் பிள்ளை - கோவில் வழிபாடு.
4. கவிக்கோ. அப்துல்ரகுமான் - பாருக்குள்ளே நல்ல நாடு.
5. சிற்பி பாலசுப்பிரமணியன் - காலம்.
6. கவிஞர் தாமரை - தொலைந்து போனேன்.

அலகு – II : அற இலக்கியம்: (8 மணிநேரம்)

1. ஔவையார் – கொன்றை வேந்தன் (1- 50 பாடல்கள்)

   அன்னையும் பிதாவும் – புலையும் கொலையும் களவும் தவிர்

2. வேதநாயகம்பிள்ளை - நீதிநூல் – (5 பாடல்கள்)

   சின்னவோர் பொருள், கடவுளை வருந்தி, எப்புவிகளும், வைத்தவர், ஈன்றவர்

3. திருவள்ளுவர்- திருக்குறள்– பண்புடைமை, வினைத்திட்பம்

4. முன்றுறையரையனார் – பழமொழி நானூறு 5 பாடல்கள்

உணற்கு இனிய, பரந்த திறலாரை, நெடியது காண்கிலாய், இனி யாரும், உரைசான்ற

அலகு – III : சிற்றிலக்கியம்: (8 மணிநேரம்)

1. முக்கூடற் பள்ளு- 2 பாடல்கள் - சித்திரக் காலிவாலான் (நெல்வகைகள்)

   குற்றாலத் திரிகூட மால்வரை (மீன் வகைகள்)

2. நந்தி கலம்பகம்- 5 பாடல்கள்- என்னையே புகழ்ந்தேன், பதிதொறு புயல்பொழி,

   இந்தப்புவியில், அடிவிளக்கும் துகில், வானுறுமதியை

3. மதுரைச் சொக்கநாதர் தமிழ்விடு தூது –தமிழின் சிறப்பு

   பாடியருள பத்துப்பாட்டும்-விளம்பக்கேள்.

அலகு – IV : சிறுகதை: **(8 மணிநேரம்)**

1.	மகாமசானம்	–	புதுமைப்பித்தன்
2.	அப்பாவின் வேஷ்டி	-	பிரபஞ்சன்
3.	அந்நியர்கள்	-	ஆர். சூடாமணி
4.	இந்நாட்டுமன்னர்	–	நாஞ்சில்நாடன்

அலகு – V : மொழிப்பயிற்சி: (6 மணிநேரம்)

1. பொருத்தமான தமிழ்ச் சொற்களைப் பயன்படுத்துதல்
2. செய்யுள் பொருளுணர் திறன்
3. மொழிபெயர்ப்புப் பயிற்சிகள்
4. கடிதங்கள் மற்றும் விண்ணப்பங்கள் எழுதுதல்

பாட நூல்: கற்பகச்சோலை – தமிழ் ஏடு.

கற்பகம் உயர்கல்வி கலைக்கழகத் தமிழ்த்துறை வெளியீடு.

Semester – I

19CSU101 PROGRAMMING FUNDAMENTALS USING C / C++ 5H – 5C

Instruction Hours / week: L: 5 T: 0 P: 0 **Marks:** Internal : **40** External : **60** Total: **100**
End Semester Exam : 3 Hours

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, putcharetc), Formatted and Console I/O (printf(), scanf(), cin, cout), Using Basic Header Files (stdio.h, iostream.h, conio.hetc).

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 access 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.) McGraw Hill, New Delhi.
2. Bjarne Stroustrup. (2013). The C++ Programming Language(4th ed.). Addison-Wesley, New Delhi.
3. Bjarne Stroustroup. (2014). Programming, Principles and Practice using C++(2nd edAddison-Wesley, New Delhi.
4. Balaguruswamy, E. (2008). Object Oriented Programming with C++. Tata McGraw-Hill Education, New Delhi.
5. Paul Deitel., & Harvey Deitel. (2011). C++ How to Program (8th ed.). Prentice Hall, New Delhi.
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/
6. www.cppreference.com/

Semester – I

19CSU102 COMPUTER SYSTEM ARCHITECTURE 4H – 4C

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

End Semester Exam: 3 Hours

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. M.Mano. (1992). Computer System Architecture. Pearson Education.
2. Dos Reis, A. J. (2009). Assembly Language and Computer Architecture using C++ and JAVA. Course Technology
3. Stallings, W. (2010). Computer Organization and Architecture Designing for Performance (8th ed.) Prentice Hall of India, New Delhi.
4. Mano, M.M. (2013). Digital Design. Pearson Education Asia, New Delhi.
5. Carl Hamacher. (2012). Computer Organization (5th ed.). McGrawHill, New Delhi.

WEB SITES

1. https://www.studytonight.com/computer-architecture/architecture-of-computer-system
2. https://www.geeksforgeeks.org/digital-electronics-logic-design-tutorials/
3. https://www.researchgate.net/publication/281003256_Computer_System_Architecture
4. https://www.studytonight.com/computer-architecture/input-output-organisation
5. computersciencewiki.org/index.php/Architecture_of_the_central_processing_unit_(CPU)

Semester – I

19CSU103 COMPUTER FUNDAMENTALS 4H – 4C

Instruction Hours / week: L: 4 T: 0 P: 0 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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

WEB SITES

1. https://www.tutorialspoint.com/computer_fundamentals/
2. https://www.javatpoint.com/computer-fundamentals-tutorial
3.https://www.researchgate.net/publication/258339295_FUNDAMENTALS_OF_COMPUTER_STUDIES
4. https://www.oreilly.com/library/view/computer-fundamentals/9788131733097/

Semester – I

19CSU111 PROGRAMMING FUNDAMENTALS USING C / C++ - PRACTICAL **4H – 2C**

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

End Semester Exam: 3 Hours

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

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

****
****

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

    i) Print the even-valued elements
    ii) Print the odd-valued elements
    iii) Calculate and print the sum and average of the elements of array

    iv) Print the maximum and minimum element of array
    v) Remove the duplicates from the array
    vi) Print the array in reverse order

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

11. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
12. Write a program that swaps two numbers using pointers.
13. Write a program in which a function is passed address of two variables and then alter its contents.
14. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.
15. Write a program to find sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions or new operator.
16. Write a menu driven program to perform following operations on strings:

    a) Show address of each character in string
    b) Concatenate two strings without using strcat function.
    c) Concatenate two strings using strcat function.
    d) Compare two strings
    e) Calculate length of the string (use pointers)
    f) Convert all lowercase characters to uppercase
    g) Convert all uppercase characters to lowercase
    h) Calculate number of vowels
    i) Reverse the string

17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
18. WAP to display Fibonacci series (i)using recursion, (ii) using iteration
19. WAP to calculate Factorial of a number (i)using recursion, (ii) using iteration
20. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.
21. Create Matrix class using templates. Write a menu-driven program to perform following Matrix operations (2-D array implementation):

a) Sum b) Difference c) Product d) Transpose

22. Create the Person class. Create some objects of this class (by taking information from the user). Inherit the class Person to create two classes Teacher and Student class. Maintain the respective information in the classes and create, display and delete objects of these two classes (Use Runtime Polymorphism).
23. Create a class Triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator.

24. Create a class Box containing length, breath and height. Include following methods in it:
    a) Calculate surface Area
    b) Calculate Volume
    c) Increment, Overload ++ operator (both prefix & postfix)
    d) Decrement, Overload -- operator (both prefix & postfix)
    e) Overload operator == (to check equality of two boxes), as a friend function
    f) Overload Assignment operator
    g) Check if it is a Cube or cuboid

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

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

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

Roll No. Name Marks

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

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

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.

Semester – I

19CSU112 COMPUTER SYSTEM ARCHITECTURE - PRACTICAL 3H – 2C

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

End Semester Exam: 3 Hours

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

Semester – I

19CSU113 COMPUTER FUNDAMENTALS - PRACTICAL 3H – 2C

Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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.

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

Add a column Region (values: S, N, N,S,S,S) between the Salesperson and Dolls columns to the given table Sort your table data by Region and within Region by Salesperson in ascending order: In this exercise, you will add a new row to your table, place the word Total at the bottom of the Salesperson column, and sum the Dolls, Trucks, and Puzzles columns.

9. Wrapping of text around the image.

10. Following features of menu option must be covered

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

MS Excel

1. Enter the Following data in Excel Sheet

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

TOTAL
AVERAGE

(a) Apply Formatting as follow:

i. Title in TIMES NEW ROMAN

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

(b) Calculate State and Qtr. Total
(c) Calculate Average for each quarter
(d) Calculate Amount = Rate * Total .

2. Given the following worksheet

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

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

If Marks	Then Grade
>= 80	A+
>= 60 < 80	A
>= 50 < 60	B
< 50	F

3. Given the following worksheet

	A	**B**	**C**	**D**	**E**	**F**	
1	**Salesman**		**Sales in (Rs.)**				
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 > 20000 and < 25000 > 25000 and < 30000 > 30000 and < 35000 >= 35000	0% of sales 4% of sales 5.5% of sales 8% of sales 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 Interest and time

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

(*a*) Calculate total sale year wise.

(*b*) Calculate the net sale made by each salesman

(*c*) Calculate the maximum sale made by the salesman

(d) Calculate the commission for each salesman under the condition.

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

(*ii*) Otherwise give 2% commission.

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

(*f*) Draw a pie graph representing the sale made by 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.

a) Calculate Monthly total.

b) Surplus = Monthly income - Monthly total.

c) What would be total surplus if monthly income is 1500.

d) How much does telephone expense for March differ from quarter average.
e) Create a 3D column graph for telephone and utilities. (f) Create a pie chart for monthly expenses.

9. Enter the following data in Excel Sheet

TOTAL REVENUE EARNED FOR SAM'S BOOKSTALL

Publisher name	1997	1998	1999	2000	total
A	Rs.1000.00	Rs.1100.00	Rs.1300.00	Rs.800.00	
B	Rs.1500.00	Rs.700.00	Rs.1000.00	Rs.2000.00	
C	Rs.700.00	Rs.900.00	Rs.1500.00	Rs.600.00	
D	Rs.1200.00	Rs.500.00	Rs.200.00	Rs.1100.00	
E	Rs.800.00	Rs.1000.00	Rs.3000.00	Rs.560.00	

(a) Compute the total revenue earned.
(b) Plot the line chart to compare the revenue of all publisher for 4 years.
(c) Chart Title should be _Total Revenue of sam's Bookstall (1997-2000)'
(d) 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.

Semester – I

19AEC101 ENVIRONMENTAL STUDIES 3H – 3C

Instruction Hours / week: L: 3 T: 0 P: 0 **Marks:** Internal : **40** External : **60** Total: **100**
End Semester Exam : 3 Hours

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 – INTRODUCTION - ENVIRONMENTAL STUDIES & ECOSYSTEMS

Environment Definition, Scope and importance; Ecosystem, Structure and functions of ecosystem. Energy flow, Food chains and food webs, Ecological succession. Classification of ecosystem. Forest ecosystem, Grassland Ecosystem, Desert ecosystem, Aquatic ecosystems (ponds, streams, lakes, rivers, oceans, estuaries).

Unit II - NATURAL RESOURCES - RENEWABLE AND NON-RENEWABLE RESOURCES

Natural resources - Renewable and Non – Renewable resources. Land resources and land use change, Land degradation, soil erosion and desertification. Forest resources - Deforestation: Causes and impacts due to mining, dam building on environment, forests, biodiversity and tribal populations. Water resources - Use and over-exploitation of surface and ground water, floods, droughts, conflicts over water. Use of alternate energy sources, growing energy needs, case studies. Role of an individual in conservation of natural resources. Equitable use of resources for sustainable lifestyles.

Unit III - BIODIVERSITY AND ITS CONSERVATION

Levels of biological diversity - genetic, species and ecosystem diversity. Ecosystem and biodiversity services: Ecological, economic, social, ethical, aesthetic and Informational value. Bio-geographical classification of India. Biodiversity patterns (global, National and local levels). Hot-spots of biodiversity. India as a mega-diversity nation. Endangered and endemic species of India. Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts. 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, Noise pollution. Nuclear hazards and human health risks. Solid waste management and control measures of urban and industrial wastes. Role of an individual in prevention of pollution. Case studies.

Unit V - SOCIAL ISSUES AND THE ENVIRONMENT

Concept of sustainability and sustainable development. Water conservation - Rain water harvesting, watershed management. Climate change, global warming, ozone layer depletion, acid rain and its impacts on human communities and agriculture. Environment Laws (Environment Protection Act, Air Act, Water Act, Wildlife Protection Act, Forest Conservation Act). International agreements (Montreal and Kyoto protocols). Resettlement and rehabilitation of project affected persons. Disaster management (floods, earthquake, cyclones and landslides). Environmental Movements (Chipko, Silent valley, Bishnois of Rajasthan). Environmental ethics: Role of Indian and other religions and cultures in environmental conservation. Environmental communication and public awareness, case studies (e.g., CNG vehicles in Delhi). Human population growth: Impacts on environment, human health and welfare.

Suggested Readings

1. Anonymous. 2004. A text book for Environmental Studies, University Grants Commission and Bharat Vidypeeth Institute of Environmental Education Research, New Delhi.
2. Anubha Kaushik., and Kaushik, C.P. 2004. Perspectives in Environmental Studies. New Age International Pvt. Ltd. Publications, New Delhi.
3. Arvind Kumar. 2004. A Textbook of Environmental Science. APH Publishing Corporation, New Delhi.
4. Daniel, B. Botkin., and Edward, A. Keller. 1995. Environmental Science John Wiley and Sons, Inc., New York.
5. Mishra, D.D. 2010. Fundamental Concepts in Environmental Studies. S.Chand & Company Pvt. Ltd., New Delhi.
6. Odum, E.P., Odum, H.T. and Andrews, J. 1971. Fundamentals of Ecology. Philadelphia: Saunders.
7. Rajagopalan, R. 2016. Environmental Studies: From Crisis to Cure, Oxford University Press**.**

8. Sing, J.S., Sing. S.P. and Gupta, S.R. 2014. Ecology, Environmental Science and Conservation. S. Chand & Publishing Company, New Delhi.
9. Singh, M.P., Singh, B.S., and Soma, S. Dey. 2004. Conservation of Biodiversity and Natural Resources. Daya Publishing House, New Delhi.
10. Tripathy. S.N., and Sunakar Panda. (2004). Fundamentals of Environmental Studies (2nd ed.). Vrianda Publications Private Ltd, New Delhi.
11. Verma, P.S., and Agarwal V.K. 2001. Environmental Biology (Principles of Ecology). S. Chand and Company Ltd, New Delhi.
12. Uberoi, N.K. 2005. Environmental Studies. Excel Books Publications, New Delhi.

கற்பகம்உயர்கல்விகலைக்கழகம்

தமிழ்த்துறை

பகுதி - I தமிழ்ப்பாடத்திட்டம் (2019 - 2020)

இரண்டாம்பருவம்

(இளநிலை அறிவியல் பட்டவகுப்புகளுக்குரியது)

(For I-UG Science Degree Classes) 19LSU201

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

- இந்தியகுடியுரிமைப்பணிமுதலானபோட்டித்தேர்வுகளில், விருப்பப்பாடமாகஇடம்பெறுகின்ற, ‘ தமிழ்இலக்கியவரலாறு’ குறித்தமுழுமையானஅறிமுகம்பெற்றிருத்தல்.
- கல்வெட்டியல், ஓலைச்சுவடியியல்மற்றும்தொல்லியல்சார்ந்தஆவணத்தேடலுக்குரியஆய்வுமனப்பான்மையுடன், இலக்கியங்களைஅணுகுதல்.
- தமிழின்வளர்ச்சித்துறையாகிய, ‘அறிவியல்தமிழ்’ ; ‘இணையதமிழ்’ குறித்தபன்நோக்குஅணுகுமுறையிலானஆய்வுச்சிந்தனைமேம்பாடு.
- வேலைவாய்ப்புக்குரியசுயதிறன்மேம்பாட்டுடன், படைப்பாக்கத்திறன்மேம்பாடும்பெற்றிருத்தல் .
- சமுதாயமற்றும்வாழ்வியல்மதிப்புகளைப்பேணுவதற்குக்கருவியாகஇலக்கியங்களைநாடுகின்றமனப்பான்மைவளர்ச்சி.
- மொழிபெயப்புத்துறைசார்ந்தவேலைவாய்புத்திறன்பெற்றிருத்தல்

தாள்கள்வரிசையும்தேர்வுச்செயல்திட்டமும்பகுதி-I தமிழ்

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

பகுதி - I தமிழ்ப் பாடத்திட்டம் (2019-2020)

பகுதி – I, தமிழ் **இரண்டாம் பருவம் 19LSU201**

தமிழ் இரண்டாம் தாள் 4-H,4-C

(இளநிலை அறிவியல் பட்ட வகுப்புகளுக்குரியது)

(For I-UG Science Degree Classes)

அலகு – I : பக்தி இலக்கியம் (7 மணிநேரம்)

சைவ, வைணவ இலக்கியங்கள் - தோற்றம் ,வளர்ச்சி, வரலாறு.

1. **சைவம் - (15 பாடல்கள்)** பெரியபுராணம் – திருமூலநாயனார் புராணம் -அந்தி இளம்பிறைக் கண்ணி, மற்று அவர்தாம் அணிமா, காவிரி நீர் பெருந்தீர்த்தம், அந்நிலைமைத் தானத்தை, அந்தணர்தம் சாத்தனூர், மற்றுஅதன் தன் உடம்பினை, இவன் உயிர்பெற் றெழில், பாய்த்திய பின் திருமூலராய், வெய்ய சுடர் கதிரவனும், அங்கவனும், பித்து உற்ற மையல் அன்று, இந்த நிலைமையில், ஆவடு தண்துறை, ஊன்உடம்பில், முன்னிய அப்பொருள்.
2. **வைணவம் – ஆண்டாள் நாச்சியார் திருப்பாவை: (11 பாடல்கள்):** மார்கழித்திங்கள், வையத்து வாழ்வீர்காள், ஓங்கி உலகளந்த, ஆழி மழைக்கண்ணா, மாயனை மன்னுவட மதுரை, சிற்றம் சிறுகாலே, ஒருத்தி மகனாய், மாலே மணிவண்ணா, கூடாரை வெல்லும், கறவைகள் பின்சென்று, வங்கக்கடல் கடைந்த.

அலகு – II : சங்க இலக்கியம் (14 மணிநேரம்)

சங்க இலக்கியங்கள் - அறிமுகம்

அ). எட்டுத்தொகை

நற்றிணை: கொண்டல் மாமழை – குறிஞ்சி – தலைவன் - 140

குறுந்தொகை: வாரார் ஆயினும், வரினும் –முல்லை– தலைவி - 110

ஐங்குறுநூறு : மருதம் –தோழி -வேட்கைப்பத்து: வாழிஆதன் வாழி அவனி - 6

பதிற்றுப்பத்து : சிதைந்தது மன்ற - 27

பரிபாடல்: பரிபாடல் திரட்டு-வையை –10: மலைவரை மாலை- கரை நன்னீர் 10:1-25, ஆங்க அணிநிலை மாடத்து- நீ காணும் போன்ம் 10: 41-55, இலம்படு புலவர் - எத்தினர் தொழுவே 10: 126-130.

கலித்தொகை : பாலைக்கலி- செவிலி – எறித்தரு கதிர்தாங்கி-9

அகநானூறு : அன்னை அறியினும் அறிக – தோழி - நெய்தல் - 110

புறநானூறு : யாதும் ஊரே யாவருங் கேளிர் –பொதுவியல்- 192

ஆ). பத்துப்பாட்டு: நெடுநல்வாடை - கார்காலச் சிறப்பு : வையகம் பனிப்ப -1-70

அலகு - III : காப்பியம் (6 மணிநேரம்)

(அ). சிலப்பதிகாரம்:

மங்கல வாழ்த்துப் பாடல்: (21-29)- நாக நீள் நகரொடு-கண்ணகி என்பாள் மன்னோ .

வழக்குரை காதை, (48-56) - நீர்வார் கண்ணை-புகா ரென்பதியே .

வஞ்சின மாலை: (5-34) - வன்னிமரமும் – பிறந்த பதிப் பிறந்தேன்.

நடுகற் காதை: (207-234) - அருந்திற லரசர் – மன்னவ ரேறென்,.

வாழ்த்துக்காதை: (9) - என்னேயிஃ தென்னே – மீவிசும்பிற் றோன்றுமால்.

(ஆ). மணிமேகலை:

பசியின் கொடுமை: பாத்திரம் பெற்ற காதை:

'போதி நீழல்' - 'பெருகியதன்றோ' , 'ஆற்றுநர்க்களிப்போர்' - 'நல்லறம் கண்டனை' (73-98).

சிறைக்கோட்டம் அறக்கோட்டமாக்கிய காதை: மாவண் கிள்ளிக்கு காவலன் உரைத்தவை:

'பைஞ்சேறு மெழுகாப் பசும்பொன் மண்டபத்து -
அரவோர்க் காக்கினன் அரசாள் வேந்தன்' (116-163).

அலகு – IV : கட்டுரைகள் (8 மணிநேரம்)

1. **தமிழிசை இயக்கம்** – பேரா. க. வெள்ளைவாரணர்
2. **தனிப்பாடல்கள்** - பேரா. வ. சுப. மாணிக்கனார்
3. **கலைச்சிறப்பு** - முத்தமிழ்க் காவலர் கி.ஆ.பெ. விசுவநாதம்
4. **காலத்தேடும் கல்வெட்டும்** –சொல்லின்செல்வர் ரா.பி.சேதுப்பிள்ளை
5. **ஏட்டில் இல்லாத இலக்கியம்** – ஔவை துரைசாமி

அலகு- V : மொழிப்பயிற்சி (5 மணிநேரம்)

(பேச்சுப்பயிற்சி & படைப்பாக்கப் பயிற்சி)

படைப்பிலக்கியப் பயிற்சிகள் (கதை, கவிதை, கட்டுரை, உரைநடை எழுதுதல், நேர்காணல்)
மொழிபெயர்ப்பு

பாடநூல்: கற்பகச்சோலை – தமிழ் ஏடு.

கற்பகம் உயர்கல்வி கலைக்கழகத் தமிழ்த்துறை வெளியீடு.

Part I TAMIL 2019. Science Karpagam Academy of Higher Education, Coimbatore – 21.

Semester – II

19ENU201 **ENGLISH** **4H – 4C**

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

End Semester Exam : 3 Hours

Course Objectives:

- To train students to acquire proficiency in English.
- To explore different genres of literature and learning grammar.
- To provide aesthetic pleasure through literature.
- To inculcate moral values through literature.
- To develop ethical values.
- To give basic grammar knowledge.

Course Outcome:

- Develop the knowledge of interpersonal skills.
- Establish and maintain social relationships.
- Genres of literature will give moral values of life.
- Develop communication skills in business environment
- Communication skills will get developed.
- Develop to have 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. Telephone Conversation- Wole Soyinka
3. A River - A.K. Ramanujan

UNIT - III : SHORT STORIES

1. Rapunzel - Brothers Grimm
2. The Ant and The Grasshopper- W. Somerset Maugham
3. The Nightingale and the Rose - Oscar Wilde.

UNIT - IV: Drama

1. The Merchant of Venice- Act 4-Scence 1
2. The Death Trap- Saki

UNIT - V: Grammar and Composition

GRAMMAR :

1. Tenses

2. Articles
3. Auxiliaries (Primary and Modal)
4. Tag Questions

Composition:

1. Reading to Comprehend
2. Letter Writing
3. Resume Writing
4. General Essay

Prescribed Text:

1. Reminisce, Published by the Department of English, Karpagam Academy of Higher Education.

Suggested Reading:

1. Hewings Martin,1999 Advanced English Grammar, Cambridge University Press

Semester – II

19CSU201 **PROGRAMMING IN JAVA** **4H – 4C**

Instruction Hours / week: L: 4 T: 0 P: 0 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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

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 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.). . Prentice Hall, New Delhi.
5. Cay, S. Horstmann., Gary Cornell. (2013). Core Java 2 Volume 2 - Advanced Features(9th ed.). Printice Hall, New Delhi.
6. Bruce Eckel. (2002). Thinking in Java (3rd ed.). PHI, New Delhi
7. Balaguruswamy, E. (2009). Programming with Java (4th ed.). McGraw Hill, New Delhi.
8. Paul Deitel., & Harvey Deitel. (2011). Java: How to Program (10th ed.). Prentice Hall, New Delhi.
9. David, J. Eck. (2009). Introduction to Programming Using Java. CreateSpace Independent Publishing Platform, New Delhi.
10. John , R. Hubbard. (2004). Programming with JAVA, Schaum's Series, (2nd ed.).

WEB SITES

1. java.sun.com/docs/books/tutorial/
2. www.java.net/

Semester – II

19CSU202 **DISCRETE STRUCTURES** **4H – 4C**

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

End Semester Exam: 3 Hours

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.

UNIT II

Pigeonhole principle, Permutation and Combination, Mathematical Induction, Principle of inclusion and Exclusion.

UNIT III

Recurrences: Recurrence relations, generating functions, linear recurrence relations with constant coefficients and their solution.

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

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

SUGGESTED READINGS

1. Kenneth Rosen.(2011). Discrete Mathematics and Its Applications (7thed.), McGraw Hill Company, New Delhi.

2. Tremblay ,J .P., and Manohar R., (2001). Discrete Mathematical Structures with Applications to Computer Science(1st ed.), McGraw-Hill Book Company, New Delhi.

3. Coremen,T.H.,Leiserson C.E. , & R. L. Rivest.(2009). Introduction to algorithms, (3rd ed.), Prentice Hall on India, New Delhi.

4. Albertson, M.O.,& Hutchinson,J.P., (1988). Discrete Mathematics with Algorithms, John Wiley Publication, New Delhi.

5. Hein,J.L., (2010). Discrete Structures, Logic, and Computability (3rd ed.), Jones and Bartlett Publishers, New Delhi.
.

6. Hunter,D.J.(2016). Essentials of Discrete Mathematics(3rd ed.). Jones and Bartlett Publishers, New Delhi.

Semester – II

19CSU203 COMPUTER NETWORKS AND INTERNET TECHNOLOGIES 4H – 4C

Instruction Hours / week: L: 4 T: 0 P: 0 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

Course Objectives

- To study the basics of Computer Networks.
- To learn various transmission media.
- To understand the topologies of networks, layered architecture (OSI and TCP/IP) and protocol suites.
- To understand the principles of creating an effective web page.
- To develop skills in analyzing the usability of a website.
- To learn the language of HTML and JavaScript.

Course Outcomes (COs)

1. Independently understand basic computer network technology.
2. Understand and explain Data Communications System and its components.
3. Identify the different types of network topologies and protocols.
4. Enumerate the layers of the OSI model and TCP/IP.
5. Gain the skills and project-based experience needed for entry into web design and development careers.
6. Select and apply markup languages for processing, identifying, and presenting of information in web pages.

Unit I - COMPUTER NETWORKS

Introduction to computer network, data communication, components of data communication, data transmission mode, data communication measurement, LAN, MAN, WAN, wireless LAN, internet, intranet, extranet. **Network Models**: Client/ server network and Peer-to-peer network, OSI, TCP/IP, 8L layers and functionalities.

Unit II - TRANSMISSION MEDIA AND LAN TOPOLOGIES

Introduction, Guided Media: Twisted pair, Coaxial cable, 4L Optical fiber. Unguided media: Microwave, Radio frequency propagation, Satellite. **LAN Topologies**: Ring, bus, star, mesh and tree topologies. Network Devices: NIC, repeaters, hub, bridge, switch, gateway and router.

Unit III - INTERNET TERMS AND APPLICATIONS

Internet Terms: Web page, Home page, website, internet browsers, URL, Hypertext, 2L ISP, Web server, download and upload, online and offline. **Internet Applications:** www, telnet, ftp, e-mail, social networks, search engines, 6L Video Conferencing, e-Commerce, m-Commerce, VOIP, blogs.

Unit IV - INTRODUCTION TO WEB DESIGN

Introduction to hypertext markup language (html) 16L Document type definition, creating web pages, lists, hyperlinks, tables, web forms, inserting images, frames, hosting options and domain name registration. **Customized Features**: Cascading style sheet (css) for text formatting and other manipulations.

Unit V - JAVASCRIPT FUNDAMENTALS

Data types and variables, functions, methods and events, 14L controlling program flow, JavaScript object model, built-in objects and operators.

SUGGESTED READINGS

1. Larry L.Peterson & Bruce S.Davie (2011). Computer Networks A System Approach, Morgan Kaufmann Publishers.
2. Andrew S. Tanenbaum, David J. Wetherall Computer Networks (5th Edition),PHI, 2010
3. B. A. Forouzan, Data Communication and Networking , TMH,2003.
4. D.R. Brooks, An Introduction to HTML and Javascript for Scientists and Engineers, Springer
5. W. Willard,2009
6. HTML A Beginner's Guide, Tata McGraw-Hill Education, 2009.
7. J. A. Ramalho, Learn Advanced HTML 4.0 with DHTML, BPB publications.

WEBSITES

1. https://developer.mozilla.org/en-US/docs/Web
2. https://www.w3schools.com
3. http://en.wikipedia.org/wiki/script_language
4. https://css-tricks.com

Semester – II

19CSU211 PROGRAMMING IN JAVA - PRACTICAL 4H – 2C

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

End Semester Exam: 3 Hours

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 StringBufferclasss like setCharAt(, setLength(), append(), insert(), concat()and equals().
9. Write a program to create a ―distance‖ class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer
10. Modify the ―distance‖ class by creating constructor for assigning values (feet and inches) to the distance object. Create another object and assign second object as reference

variable to another object reference variable. Further create a third object which is a clone of the first object.

11. Write a program to show that during function overloading, if no matching argument is found, then java will apply automatic type conversions(from lower to higher data type)
12. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword
13. Write a program to show the use of static functions and to pass variable length arguments in a function.
14. Write a program to demonstrate the concept of boxing and unboxing.
15. Create a multi-file program where in one file a string message is taken as input from the user and the function to display the message on the screen is given in another file (make use of Scanner package in this program).
16. Write a program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate 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.

Semester – II

19CSU212 DISCRETE STRUCTURES - PRACTICAL 3H – 2C

Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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) ………+1/(n(n+1)) = n/(n+1)
4. Write a C Program to perform the sum = 1+ (1+2) + (1+2+3) + … + (1+2…+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) $P\wedge Q\wedge \rceil R$ (ii) $P\wedge \rceil Q\wedge R$ (iii) $P\wedge Q\wedge \rceil R$
10. Write a C Program to prove De – Morgan’s law.

Semester – II

19CSU213 COMPUTER NETWORKS AND INTERNET TECHNOLOGIES - PRACTICAL 3H – 2C

Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

Course Objectives

- To study the basics of Computer Networks.
- To learn various transmission media.
- To understand the topologies of networks, layered architecture (OSI and TCP/IP) and protocol suites.
- To understand the principles of creating an effective web page.
- To develop skills in analyzing the usability of a website.
- To learn the language of HTML and JavaScript.

Course Outcomes (COs)

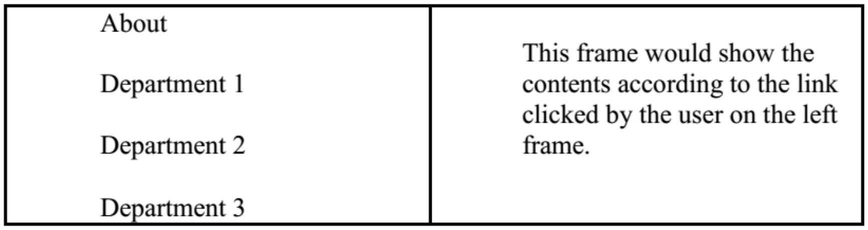
1. Independently understand basic computer network technology.
2. Understand and explain Data Communications System and its components.
3. Identify the different types of network topologies and protocols.
4. Enumerate the layers of the OSI model and TCP/IP.
5. Gain the skills and project-based experience needed for entry into web design and development careers.
6. Select and apply markup languages for processing, identifying, and presenting of information in web pages.

List of Programs

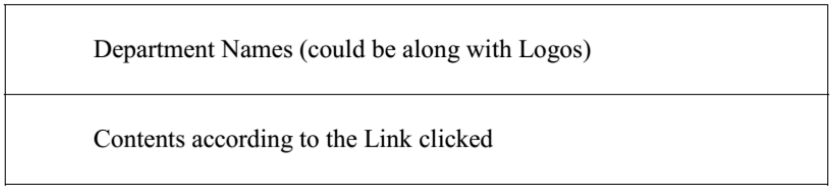
1. Create HTML document with following formatting – Bold, Italics, Underline, Colors, Headings, Title, Font and Font Width, Background, Paragraph, Line Brakes, Horizontal Line, Blinking text as well as marquee text.
2. Create HTML document with Ordered and Unordered lists, Inserting Images, Internal and External linking
3. Create HTML document with Table

<table>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="3"></td><td colspan="2" rowspan="2"></td></tr>
<tr><td colspan="3"></td></tr>
</table>

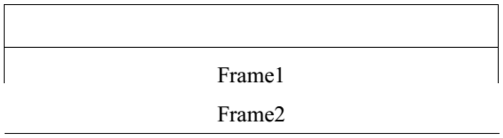
4. Create Form with Input Type, Select and Text Area in HTML.
5. Create an HTML containing Roll No., student‘s name and Grades in a tabular form.
6. Create an HTML document (having two frames) which will appear as follows

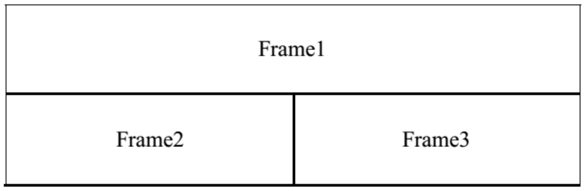

About
Department 1
Department 2
Department 3
This frame would show the contents according to the link clicked by the user on the left frame.

7. Create an HTML document containing horizontal frames as follows


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

8. Create a website of 6 – 7 pages with different effects as mentioned in above problems.
9. Create HTML documents (having multiple frames) in the following three formats





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

    V. Text Box
    VI. Option/radio buttons
    VII. Check boxes
    VIII. Reset and Submit buttons

List of Practical using Javascript : Create event driven program for following:

11. Print a table of numbers from 5 to 15 and their squares and cubes using alert.
12. Print the largest of three numbers. 81
13. Find the factorial of a number n.
14. Enter a list of positive numbers terminated by Zero. Find the sum and average of these numbers.
15. A person deposits Rs 1000 in a fixed account yielding 5% interest. Compute the amount in the account at the end of each year for n years.
16. Read n numbers. Count the number of negative numbers, positive numbers and zeros in the list.

Semester – III

19CSU301 **DATA STRUCTURES** **4H – 4C**

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

End Semester Exam: 3 Hours

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 collusion 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., & Yedidyah 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., & Yedidyah Langsam. (2003). Data Structures Using Java. New Delhi: PHI.
8. Robert Lafore. (2003). Data Structures and Algorithms in Java(2nd 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

1. http://en.wikipedia.org/wiki/Data_structure
2. http://www.cs.sunysb.edu/~skiena/214/lectures/
3. www.amazon.com/Teach-Yourself-Structures-Algorithms

Semester – III

19CSU302 **OPERATING SYSTEMS** **4H – 4C**

Instruction Hours / week: L: 4 T: 0 P: 0 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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 access Authorization.

SUGGESTED READINGS

1. Silberschatz, A ., Galvin, P.B. , & Gagne, G. (2012). Operating Systems Concepts, 8th ed.. New Delhi: John Wiley Publications.
2. Tanenbaum, A.S. (2007).Modern Operating Systems (3rd 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/
2. www.clarkson.edu/~jnm/cs644
3. pages.cs.wisc.edu/~remzi/Classes/736/Fall2002/

Semester – III

19CSU303 **COMPUTER NETWORKS** **4H – 4C**

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

End Semester Exam: 3 Hours

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

Unit II - NETWORKS SWITCHING TECHNIQUES AND ACCESS MECHANISMS

Circuit switching; packetswitching - 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. (2012). Data Communications and Networking. 4th edition. New Delhi: THM.
2. Tanenbaum, A. S. (2002). Computer Networks. 4th edition. New Delhi: PHI.

WEB SITES

1. en.wikipedia.org/wiki/Internet_protocol_suite
2. http://docwiki.cisco.com/wiki/Introduction_to_WAN_Technologies
3. www.yale.edu/pclt/COMM/**TCPIP**.HTM
4. www.w3schools.com/**tcpip**/default.asp

Semester – III

19CSU304A ANDROID PROGRAMMING 3H – 3C

Instruction Hours / week: L: 3 T: 0 P: 0 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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.

WEB SITES

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.html
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

Semester – III

19CSU304B PROGRAMMING IN VISUAL BASIC /GAMBAS 3H – 3C

Instruction Hours / week: L: 3 T: 0 P: 0 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

Course Objectives

- To identify the differences between the procedural languages and event driven languages.
- To introduce computer programming using the VISUAL BASIC programming language.
- To demonstrate knowledge of programming terminology and how applied using Visual Basic (e.g., variables, selection statements, repetition statements, etc.)
- To develop a Graphical User Interface (GUI) based on problem description
- To develop and debug applications using Visual Basic
- To Emphasis on event-driven programming methods, including creating and manipulating objects, classes, and using object-oriented tools such as the class debugger.

Course Outcomes(COs)

1. Construct appropriate user interfaces for simple programs, and design systems with minimal complexity and maximal functionality.
2. Understand computer programming using the VISUAL BASIC programming language.
3. Demonstrate knowledge of programming terminology and how applied using Visual Basic (e.g., variables, selection statements, repetition statements, etc.)
4. Develop a Graphical User Interface (GUI) based on problem description
5. Develop and debug applications using Visual Basic
6. Emphasize on event-driven programming methods, including creating and manipulating objects, classes, and using object-oriented tools such as the class debugger.

Unit I - GUI ENVIRONMENT

Introduction to graphical user interface (GUI), programming language (procedural, object oriented, event driven), the GUI environment, compiling, debugging, and running the programs.
Controls : Introduction to controls textboxes, frames, check boxes, option buttons, images, setting borders and styles, the shape control, the line control, working with multiple controls and their properties, designing the user interface, keyboard access, tab controls, default & cancel property, coding for controls.

Unit II – OPERATIONS

Data types, constants, named & intrinsic, declaring variables, scope of variables, val function, arithmetic operations, formatting data.
Decision Making : If statement, comparing strings, compound conditions (and, or, not), nested if statements, case structure, using if statements with option buttons & check boxes, displaying message in message box, testing whether input is valid or not.

Unit III -MODULAR PROGRAMMING

Menus, sub-procedures and sub-functions defining/creating and modifying a menu, using common dialog box, creating a new subprocedure, passing variables to procedures, passing argument by value or by reference, writing a function/ procedure.

Unit IV- FORMS HANDLING

Multiple forms creating, adding, removing forms in project, hide, show method, load, unload statement, me keyword, referring to objects on a different forms

Iteration Handling: Do/loops, for/next loops, using msgbox function, using string Function

Unit V- ARRAYS AND GROUPED DATA CONTROL

Arrays - 1-dimension arrays, initializing and array using for each, user-defined data types, accessing information with user-defined data types, using list boxes with array, two dimensional arrays. lists, loops and printing list boxes & combo boxes, filling the list using property window / add item method, clear method, list box properties, removing an item from a list, list box/ combo box operations.

Database Connectivity: Database connectivity of forms with back end tool like mysql populating the data in text boxes, list boxes etc. searching of data in database using forms. Updating/ editing of data based on a criterion.

SUGGESTED READINGS

1. Programming in Visual Basic 6.0 by Julia Case Bradley, Anita C. Millispangh. 2014. Tata Mcgraw Hill Edition.

WEB SITES

1. https://www.tutorialspoint.com/vb.net
2. howtostartprogramming.com/vb-net/
3. https://www.vbtutor.net/lesson1.html
4. gambas.sourceforge.net/

Semester – III

19CSU311 **DATA STRUCTURES - PRACTICAL** **4H – 2C**

Instruction Hours / week: L: 0 T: 0 P: 4 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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.

Semester – III

19CSU312 OPERATING SYSTEMS - PRACTICAL 4H – 2C

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

End Semester Exam: 3 Hours

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.

Semester – III

19CSU313 COMPUTER NETWORKS - PRACTICAL 4H – 2C

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

End Semester Exam: 3 Hours

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.

Semester – III

19CSU314A ANDROID PROGRAMMING - PRACTICAL 3H – 1C

Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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 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.

Semester – III

19CSU314B PROGRAMMING IN VISUAL BASIC/GAMBAS – PRACTICAL 3H – 1C

Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

Course Objectives

- To identify the differences between the procedural languages and event driven languages.
- To introduce computer programming using the VISUAL BASIC programming language.
- To demonstrate knowledge of programming terminology and how applied using Visual Basic (e.g., variables, selection statements, repetition statements, etc.)
- To develop a Graphical User Interface (GUI) based on problem description
- To develop and debug applications using Visual Basic
- To Emphasis on event-driven programming methods, including creating and manipulating objects, classes, and using object-oriented tools such as the class debugger.

Course Outcomes(COs)

1. Construct appropriate user interfaces for simple programs, and design systems with minimal complexity and maximal functionality.
2. Understand computer programming using the VISUAL BASIC programming language.
3. Demonstrate knowledge of programming terminology and how applied using Visual Basic (e.g., variables, selection statements, repetition statements, etc.)
4. Develop a Graphical User Interface (GUI) based on problem description
5. Develop and debug applications using Visual Basic
6. Emphasize on event-driven programming methods, including creating and manipulating objects, classes, and using object-oriented tools such as the class debugger.

List of Programs

1. Print a table of numbers from 5 to 15 and their squares and Cubes.
2. Print the largest of three numbers.
3. Find the factional of a number n.
4. Enter a list of positive numbers terminated by zero. Find the sum and average of these numbers.
5. A person deposits Rs. 1000 in a fixed account yielding 5% interest. Complete the amount in the account at the end of each year for n years.
6. Read n numbers. Count the number of negative numbers, positive numbers and zeros in the list.
7. Read n numbers. Count the number of negative numbers, positive numbers and zeroes in the list.use arrays.
8. Read a single dimension array. Find the sum and average of these numbers.
9. Read a two dimension array. Find the sum of two 2D Array.
10. Create a database Employee and Make a form in VB 6.0 to allow data entry to Employee Form with the following command buttons:

Employee Form

```
Employee Name: NEXT
Employee Id:
Date of Joining:
Designation:
Department:
Address:
```

Semester – IV

19CSU401 DESIGN AND ANALYSIS OF ALGORITHMS 4H – 4C

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

End Semester Exam: 3 Hours

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 edition. New Delhi: PHI.
2. Sarabasse., Gelder, A.V. (1999). Computer Algorithm – Introduction to Design and Analysis. 3rd edition. New Delhi: Pearson

WEB SITES

1. https://www.tutorialspoint.com/design_and_analysis_of_algorithms

2. https://www.javatpoint.com/daa-tutorial

3. www.vssut.ac.in/lecture_notes/lecture1428551222.pdf

Semester – IV

19CSU402 SOFTWARE ENGINEERING 4H – 4C

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

End Semester Exam: 3 Hours

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 EngineeringProcess- Requirement Analysis and Modeling Techniques- FlowOriented 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. (2014). Software Engineering: A Practitioner‘s Approach. 7th edition. New Delhi: McGraw-Hill.
2. Jalote, P. (2012). An Integrated Approach to Software Engineering. 2nd edition. New Delhi: New Age International Publishers.
3. Aggarwal, K.K., & Singh, Y. (2012). Software Engineering. 2nd edition. New Delhi: New Age International Publishers.
4. Sommerville, I. (2006). Software Engineering. 8th edition. New Delhi: Addison Wesley.
5. Bell, D. (2005). Software Engineering for Students. 4th edition. New Delhi: Addison-Wesley.
6. Mall, R. (2004). Fundamentals of Software Engineering. 2nd edition. New Delhi: Prentice-Hall of India.

WEB SITES

1. 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/

Semester – IV

19CSU403 DATA BASE MANAGEMENT SYSTEMS 4H – 4C

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

End Semester Exam: 3 Hours

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 - SQL QUERIES

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

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.

SUGGESTED READINGS

1. Elmasri, R., & Navathe, S.B. (2012). Fundamentals of Database Systems. 6th edition. New Delhi: Pearson Education,.
2. Ramakrishanan, R., & Gehrke, J. (2010). Database Management Systems. 3rd edition. New Delhi: McGraw-Hill.
3. Silberschatz, A., Korth, H.F., & Sudarshan, S. (2010).Database System Concepts. 6th edition. New Delhi: McGraw-Hill
4. Elmasri, R., & Navathe, S.B. (2013). Database Systems Models, Languages, Design and application Programming. 6th edition. 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
3. www.compinfo-center.com/apps/rdbms.html

Semester – IV

19CSU404A HTML PROGRAMMING 3H – 3C

Instruction Hours / week: L: 3 T: 0 P: 0 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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 TO 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. (2016). 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/
2. alexle.net/archives/category/web-technolgy
3. jmarshall.com/easy/
4. www.php.net/
5. en.wikipedia.org/wiki/PHP
6. www.w3schools.com/PHP/DEfaULT.asP

Semester – IV

19CSU404B XML PROGRAMMING 3H – 3C

Instruction Hours / week: L: 3 T: 0 P: 0 Marks: Internal: 40 External: 60 Total: 100

End Semester Exam: 3 Hours

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. (2012). XML in action web technology. Microsoft Press.
2. Michael, J. Young. (2014). Step by Step XML. Microsoft Press.

WEB SITES

1. https://epdf.tips/xml-step-by-step.html
2. https://trove.nla.gov.au/work/34083699
3. https://store.testing.jaga-me.com/xml_step_by_step_dv_dlt_fundamentals.pdf

Semester – IV

19CSU411 DESIGN AND ANALYSIS OF ALGORITHMS - PRACTICAL **4H – 2C**

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

End Semester Exam: 3 Hours

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 nlogn.

Semester – IV

19CSU412 SOFTWARE ENGINEERING - PRACTICAL 4H – 2C

Instruction Hours / week: L: 0 T: 0 P: 4 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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.	• Problem Statement • Process Model
2.	Requirement Analysis: • Creating a Data Flow • Data Dictionary, Use Cases
3.	Project Management: • Computing FP • Effort • Schedule, Risk Table, Timeline chart
4.	Design Engineering: • Architectural Design • Data Design, Component Level Design
5.	Testing: • Basis Path Testing

Sample Projects:

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

Semester – IV

19CSU413 DATA BASE MANAGEMENT SYSTEMS - PRACTICAL 4H – 2C

Instruction Hours / week: L: 0 T: 0 P: 4 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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.

Semester – IV

19CSU414A HTML PROGRAMMING - PRACTICAL 3H – 1C

Instruction Hours / week: L: 0 T: 0 P: 3 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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.

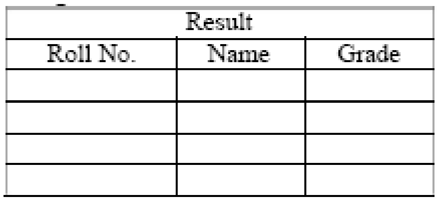
List of Programs

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

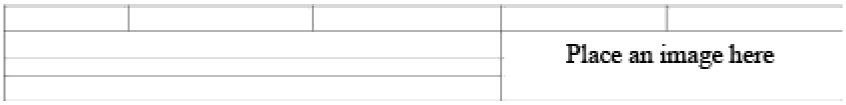
3 Create an HTML document which implements Internal linking as well as External linking.

4 Create a table using HTML which consists of columns for Roll No., Student‘s name and grade.



Result		
Roll No.	Name	Grade

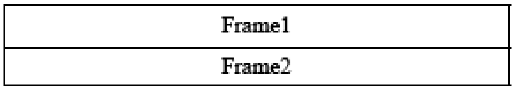
5 Create a Table with the following view:

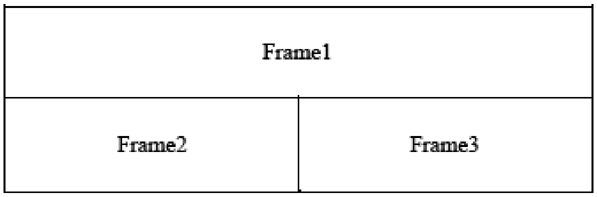


6 Create a form using HTML which has the following types of controls:

I. Text Box
II. Option/radio buttons
III. Check boxes
IV. Reset and Submit buttons

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





Semester – IV

19CSU414B XML PROGRAMMING - PRACTICAL 3H – 1C

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

End Semester Exam: 3 Hours

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.

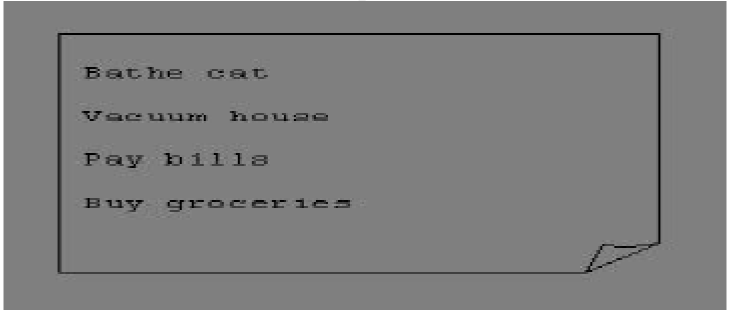
Course Outcomes(COs)

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.



Bathe cat

Vacuum house

Pay bills

Buy groceries

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>
```

Semester – V

19CSU501A **CLOUD COMPUTING** **4H – 4C**

Instruction Hours / week: L: 4 T: 0 P: 0 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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. (2014). 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. wikipedia.org/wiki/Cloud_computing
2. www.ibm.com/cloud-computing/in/en/
3. www.oracle.com/CloudComputing
4. www.microsoft.com/en-us/cloud/default.aspx

Semester – V

19CSU501B SOFTWARE TESTING 4H – 4C

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

End Semester Exam: 3 Hours

Course Objective:

- To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
- To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
- To learn how to planning a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report.
- To expose the advanced software testing topics, such as object-oriented software testing methods, and component-based software testing issues, challenges, and solutions.
- To gain software testing experience by applying software testing knowledge and methods to practice-oriented software testing projects.
- To gain the techniques and skills on how to use modern software testing tools to support software testing projects

Course Outcomes(COs)

1. Have the ability to understand the fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods
2. Have an ability to design and conduct a software test process for a software testing project.
3. Have an ability to identify the needs of software test automation, and define and develop a test tool to support test automation.
4. Have an ability understand and identify various software testing problems, and solve these problems by designing and selecting software test models, criteria, strategies, and methods.
5. Have an ability to use various communication methods and skills to communicate with their teammates to conduct their practice-oriented software testing projects.
6. Have an ability to use software testing methods and modern software testing tools for their testing projects.

Unit I – INTRODUCTION

Strategic Approach to Software Testing, Test Strategies for Conventional Software, Validation Testing, System Testing.

Unit II - TESTING

Basic Terminologies, V Shaped Software Lifecycle Model, Functional Testing- Black-box Testing.

Unit III - TESTING

Boundary Value Analysis, Equivalence Class Testing, Decision Table Based Testing
Structural Testing\ White-box Testing.

Unit IV - BASIS PATH TESTING

Program Graph, DD Path graph, Cyclomatic Complexity, Graph Matrices, Control Flow Testing: Statement Coverage, Branch Coverage, Condition Coverage, Path Coverage

Unit V - OBJECT ORIENTED TESTING

Object Orientation , Path Testing, State Based Testing, Class Testing.Metrics and Models in Software Testing:Software Metrics , Categories of Metrics, Object Oreineted Metrics used in Testing.

SUGGESTED READINGS

1. Roger S. Pressman. 2009. Software Engineering: A Practitioner‘s Approach, Seventh Edition, McGraw Hill Education.
2. Yogesh Singh. 2011. Software Testing. Cambridge University Press.

WEBSITES

1. www.tutorialspoint.com/software_testing/
2. software-engineering/object-oriented-testing

Semester – V

19CSU502A INTERNET TECHNOLOGIES 4H – 4C

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

End Semester Exam: 3 Hours

Course Objectives

- To get familiar with basics of the Internet Programming.
- To acquire knowledge and skills for creation of web site considering both client and server side programming
- To gain ability to develop responsive web applications using JavaScript
- To develop advanced Java applications using JDBC
- To learn and create JSP applications
- To develop the component based applications using Java Beans

Course Outcomes(COs)

1. Familiarize with the basics of the Internet Programming.
2. Acquires knowledge and skills for creation of web site considering both client and server side programming
3. Ability to develop responsive web applications using JavaScript
4. Develop advanced Java applications using JDBC
5. Understand and create JSP applications
6. Develop the component based applications using Java Beans

Unit I - JAVA

Use of Objects, Array and ArrayList class

Unit II - JAVASCRIPT

Data types, operators, functions, control structures, events and event handling.

Unit III - JDBC

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

Unit IV - JSP

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

Unit V: Java Beans

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

SUGGESTED READINGS

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

WEBSITES

1. www.ntu.edu.sg/home/ehchua/programming/java/JSPByExample.html
2. https://docs.oracle.com/cd/E15523_01/web.1111/e13712/reference.htm

3. https://www.tutorialspoint.com/jsp/jsp_quick_guide.htm

Semester – V

19CSU502B INFORMATION SECURITY AND CYBER LAWS 4H – 4C

Instruction Hours / week: L: 4 T: 0 P: 0 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

Course Objectives

- To provides 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 four techniques for crypto-analysis symmetric and asymmetric cryptography, 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
- To understand the various Cyber laws and its sections with case studies.

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 four techniques for crypto-analysis symmetric and asymmetric cryptography, digital signature, message authentication code, hash functions and modes of encryption operations.
4. Explain common vulnerabilities in computer programs including buffer overflow Vulnerabilities time-of-check to time-of-use flaws incomplete mediation.
5. Understand the concepts of cryptographic utilities and authentication mechanisms to design secure applications
6. Understand the various Cyber laws and its sections with case studies.

Unit I - COURSE INTRODUCTION

Computer network as a threat, hardware vulnerability, software vulnerability, importance of data security.
Digital Crime: Overview of digital crime, criminology of computer crime.

Unit II - INFORMATION GATHERING TECHNIQUES

Tools of the attacker, information and cyber warfare, scanning and spoofing, password cracking, malicious software, session hijacking

Unit III - RISK ANALYSIS AND THREAT

Risk analysis, process, key principles of conventional computer security, security policies, authentication, data protection, access control, internal vs external threat, security assurance, passwords, authentication, and access control, computer forensics and incident response

Unit IV- INTRODUCTION TO CRYPTOGRAPHY AND APPLICATIONS

Important terms, Threat, Flaw, Vulnerability, Exploit, Attack, Ciphers, Codes, Caeser Cipher, Rail-Fence Cipher, Public key cryptography (Definitions only), Private key cryptography (Definition and Example)
Safety Tools and Issues : Firewalls, logging and intrusion detection systems, Windows and windows XP / NT security, Unix/Linux security, ethics of hacking and cracking

Unit V- CYBER LAWS

CYBER LAWS to be covered as per IT 2008:

- Chapter 1: Definitions 88
- Chapter 2: Digital Signature And Electronic Signature
- [Section 43] Penalty and Compensation for damage to computer, computer system, etc.
- [Section 65] Tampering with Computer Source Documents
- [Section 66 A] Punishment for sending offensive messages through communication service, etc.
- [Section 66 B] Punishments for dishonestly receiving stolen computer resource or communication device
- [Section 66C] Punishment for identity theft
- [Section 66D] Punishment for cheating by personation by using computer resource
- [Section 66E] Punishment for violation of privacy
- [Section 66F] Punishment for cyber terrorism
- [Section 67] Punishment for publishing or transmitting obscene material in electronic form
- [Section 67A] Punishment for publishing or transmitting of material containing sexually explicit act, etc. in electronic form[Section 67B] Punishment for publishing or transmitting of material depicting children in sexually explicit act, etc. in electronic form
- [Section 72] Breach of confidentiality and privacy

SUGGESTED READINGS

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

WEB SITES

1. http://www.csc.ncsu.edu/faculty/ning
2. csrc.nist.gov/publications/nistpubs/800-12/handbook.pdf
3. www2.warwick.ac.uk/fac/sci/dcs/teaching/modules/cs134/

Semester – V

19CSU503A DATA MINING 4H – 4C

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

End Semester Exam: 3 Hours

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 - LEARNING TECHNIQUES

Supervised and unsupervised learning techniques

Unit III - KNOWLEDGE DISCOVERY

Process of knowledge discovery in databases, pre-processing methods

Unit IV - DATA MINING TECHNIQUES

Association Rule Mining, classification and regression techniques, clustering

Unit V - ISSUES

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

SUGGESTED READINGS

1. Pang-Ning Tan., Michael Steinbach.,& Vipin Kumar. (2012). Introduction to Data Mining. New Delhi: Pearson Education.
2. Richard Roiger., & Michael Geatz. (2013). 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

Semester – V

19CSU503B R-PROGRAMMING 4H – 4C

Instruction Hours / week: L: 4 T: 0 P:0 Marks: Internal: 40 External: 60 Total: 100
End Semester Exam: 3 Hours

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: INTRODUCTION

History and Overview of R, Getting Started with R, Getting Help, Data Types, R Nuts and Bolts, **Getting Data In and out of R**: Reading and Writing Data, **Subsetting R objects:** Vector, Matrix, List and Data frames.

Unit II: R FUNCTIONALITIES

Operators in R, Vectorized Operations, **Date and Times in R:** Operations on date and times, **Managing Data frames with the dplyr package:** Data Frames, The dplyr package, dplyr Grammar, select( ),arrange( ), filter( ), rename( ), mutate( ),group_by( ),%>%.

Unit III: CONTROL STRUCTURES:

if_else, else_if, for loops, Nested for loop, while, repeat, next, break-Scoping Rules-**Functions in R:** lapply, tapply, split, mapply, apply-Combining Variables with the c, cbind, rbind Functions-Coding Standards in R-String Operations.

Unit IV: STATISTICAL ANALYSIS IN R

Statistical Analysis in R: Data types – Categorical – Binary – ordinal – Nominal – Continuous – Discrete – Data Dimensions – Univariate – bivariate – multivariate – Numerical Measures – Central Tendency – Mean – Median – Mode.R Packages-Debugging Tools-Simulation-R Profiler-Statistical Functions – Comparison of Samples – same groups – different groups – Independent groups - Student T Test – Dependent Test – Independent Test.

Unit V: IMPORT AND EXPORT DATA INTO R

Read CSV, Excel, SPSS, Stata, SAS Files. **Data visualization**: Base graphics system in R, Advanced R graphics: ggplot - **Reporting** – Data Preparation – Embedding R chunks – Labelling and reusing code chunks – Report Compiling – Configuring – R Packages – shiny - ggvis

SUGGESTED READINGS

1. William N. Venables and David M. Smith. An Introduction to R. 2nd Edition.Network Theory Limited.2009.
2. Norman Matloff. The Art of R Programming - A Tour of Statistical Software Design, No Starch Press.2011.

WEB SITES

1. https://www.w3schools.in/r
2. https://www.analyticsvidhya.com/blog/2016/02/complete-tutorial-learn-data-science-scratch/
3. https://www.statmethods.net/r-tutorial
4. https://www.cs.upc.edu/~robert/teaching/estadistica/rprogramming.pdf
5. https://www.tutorialkart.com/r-tutorial

Semester – V

19CSU504A **ORACLE (SQL/PL-SQL)** **3H – 3C**

Instruction Hours / week: L: 3 T: 0 P: 0 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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 edition. 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.

WEBSITES

1. https://docs.oracle.com/cd/B28359_01/appdev.111/b28370/toc.htm

2. https://docs.oracle.com/database/121/LNPLS/toc.htm

3. https://docs.oracle.com/cd/E11882_01/appdev.112/e25519.pdf

Semester – V

19CSU504B PROGRAMMING IN PYTHON 3H – 3C

Instruction Hours / week: L: 3 T: 0 P: 0 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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 edition. New Delhi: TMH.
2. Python Tutorial/Documentation. (2015). www.python.org.
3. Allen Downey., Jeffrey Elkner., & Chris Meyers. (2012). How to think like a computer scientist : learning with Python. Freely available online.

WEBSITES

1. http://docs.python.org/3/tutorial/index.html.
2. http://interactivepython.org/courselib/static/pythonds.
3. http://www.ibiblio.org/g2swap/byteofpython/read/.

Semester – V

19CSU511A CLOUD COMPUTING -PRACTICAL 4H – 2C

Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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

Semester – V

19CSU511B SOFTWARE TESTING –PRACTICAL 4H – 2C

Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

Course Objective:

- To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
- To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
- To learn how to planning a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report.
- To expose the advanced software testing topics, such as object-oriented software testing methods, and component-based software testing issues, challenges, and solutions.
- To gain software testing experience by applying software testing knowledge and methods to practice-oriented software testing projects.
- To gain the techniques and skills on how to use modern software testing tools to support software testing projects

Course Outcomes(COs)

1. Have the ability to understand the fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods
2. Have an ability to design and conduct a software test process for a software testing project.
3. Have an ability to identify the needs of software test automation, and define and develop a test tool to support test automation.
4. Have an ability understand and identify various software testing problems, and solve these problems by designing and selecting software test models, criteria, strategies, and methods.
5. Have an ability to use various communication methods and skills to communicate with their teammates to conduct their practice-oriented software testing projects.
6. Have an ability to use software testing methods and modern software testing tools for their testing projects.

Computer Lab Based on Software Testing

1. Write a program that take three inputs (a,b &c) that represent the sides of a triangle, and the output is one of the below four:
a. Not a triangle
b. Scalene triangle
c. Isosceles triangle
d. Equilateral triangle
1.1 Generate test cases using Boundary Value Analysis, Equivalence Class Partitioning and Decision Table Testing.
1.2 Generate test cases using Basis path testing.

1.3 Run code coverage tool.

2. Write a program that determines the nature of roots of a quadratic equation. Output should be one of the following:-

- Not a quadratic equation.
- Complex roots
- Real roots
- Single roots

I. Generate test cases using Boundary Value Analysis, Equivalence Class Partitioning and Decision Table Testing.
II. Generate test cases using Basis path testing.
III. Run code coverage tool

3. Write a program that checks whether the number is even or odd. Run code coverage tool and find the amount of code being covered.
4. Write a program that dynamically allocates memory to10 integers using malloc() or calloc() And

- donot free memory leading to memory leaks. Verify the same usingValgrind.
- Now, free memory using free() at the end of the program to avoid memory leaks. Verify the same using Valgrind.

5. Use LoadUI load testing tool to test the web application performance

Semester – V

19CSU512A INTERNET TECHNOLOGIES – PRACTICAL 4H – 2C

Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

Course Objectives

- To get familiar with basics of the Internet Programming.
- To acquire knowledge and skills for creation of web site considering both client and server side programming
- To gain ability to develop responsive web applications using JavaScript
- To develop advanced Java applications using JDBC
- To learn and create JSP applications
- To develop the component based applications using Java Beans

Course Outcomes(COs)

1. Familiarize with the basics of the Internet Programming.
2. Acquires knowledge and skills for creation of web site considering both client and server side programming
3. Ability to develop responsive web applications using JavaScript
4. Develop advanced Java applications using JDBC
5. Understand and create JSP applications
6. Develop the component based applications using Java Beans

Create event driven program for following:

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

Semester – V

19CSU512B INFORMATION SECURITY AND CYBER LAWS - PRACTICAL 4H – 2C

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

End Semester Exam: 3 Hours

Course Objectives

- To provides 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 four techniques for crypto-analysis symmetric and asymmetric cryptography, 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
- To understand the various Cyber laws and its sections with case studies.

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 four techniques for crypto-analysis symmetric and asymmetric cryptography, digital signature, message authentication code, hash functions and modes of encryption operations.
4. Explain common vulnerabilities in computer programs including buffer overflow Vulnerabilities time-of-check to time-of-use flaws incomplete mediation.
5. Understand the concepts of cryptographic utilities and authentication mechanisms to design secure applications
6. Understand the various Cyber laws and its sections with case studies.

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.

Semester – V

19CSU513A **DATA MINING - PRACTICAL** **4H – 2C**

Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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

LIST OF PRACTICALS

1. Use the following learning schemes, with the default settings to analyze the weather data (in weather.arff). for test options, first choose “Use training set”, then choose “Percentage split” using default 66% percentage split. Report model percent error rate.
2. Use iris dataset preprocess and classify it with j4.8 and Naive Bayes classifier. Examine the tree in the classifier output panel.
3. Using the dataset ReutersCorn – Train and ReutersGrain – Train. Classify articles using binary attributes and word count attributes.
4. Apply any two association rule based algorithm for the supermarket analysis.
5. Using weka experimenter perfrom comparison analysis of j4.8, oneR and ID3 for vote dataset.
6. Using weka experimenter perfrom comparison analysis of Naive Bayes with different datasets.
7. Apply ZeroR, OneR and j4.8, to classify the iris data in an experiment using 10 train and test runs, with 66% of the data used for 34% used for testing.
8. Using Weka Knowledge flow set up a flow to load an ARFF file (batch mode) and perform a cross-validation using j4.8 (WEKS’s C4.5 implementation).
9. Draw multiple ROC curves in the same plot window, using j4.8 and RandomForest as classifiers.
10. Use any three clustering algorithm on Vehicle data set and find best among them.

Semester – V

19CSU513B **R PROGRAMMING – PRACTICAL** **4H – 2C**

Instruction Hours / week: L: 0 T: 0 P: 4 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

Course Objectives

- 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.

Software Lab Based on R Programming

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 the following sorting algorithms: Selection sort, Insertion sort, Bubble Sort
8. Implement linear search.
9. Implement binary search.
10. Implement matrices addition, subtraction and Multiplication

Semester – V

19CSU514A **Oracle (SQL/PL-SQL) LAB - PRACTICAL** **3H – 1C**

Instruction Hours / week: L: 0 T: 0 P: 3 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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

Semester – V

19CSU514B PROGRAMMING IN PYTHON - PRACTICAL 3H – 1C

Instruction Hours / week: L: 0 T: 0 P: 3 Marks: Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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 aCSUording to the following criteria :
   Grade A: Percentage >=80
   Grade B: Percentage>=70 and <80
   Grade C: Percentage>=60 and <70
   Grade D: Percentage>=40 and <60
   Grade E: Percentage<40
3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by aCSUepting suitable input parameters from user.
4. WAP to display the first n terms of FibonaCSUi 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! - - - - - n/n!
7. WAP to calculate the sum and product of two compatible matrices.

Semester – VI

19CSU601A PHP PROGRAMMING 4H – 4C

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

End Semester Exam: 3 Hours

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 rguments, 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 Professsional.
5. David Sklar., & Adam Trachtenberg. PHP Cookbook: Solutions & Examples for PHP.

WEBSITES

1. https://www.w3schools.com/php/php_ref_overview.asp
2. https://www.php.net/manual/en/language.references.php
3. https://www.php.net/manual/en/langref.php
4. https://www.php.net/manual/en/language.references.whatdo.php

Semester – VI

19CSU601B UNIX / LINUX PROGRAMMING 4H – 4C

Instruction Hours / week: L: 4 T: 0 P: 0 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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.).

WEBSITE

1. www.sethi.org/tutorials/references_unix.html

2. www.penguintutor.com/linux/basic-shell-programming-reference

3. https://swcarpentry.github.io/shell-novice/reference/

Semester – VI

19CSU602A WEB AND E-COMMERCE TECHNOLOGIES 4H – 4C

Instruction Hours / week: L: 4 T: 0 P: 0 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

Course Objectives

- to describe Internet trading relationships including Business to Consumer, Business-to-Business, Intra-organizational.
- To analyse the impact of E-commerce on business models and strategy
- To describe the key features of Internet, Intranets and Extranets and explain how they relate to each other.
- To understand applied cryptographic technology and Web security protocols;
- To understand the necessary infrastructure and functional components to develop Ecommerce systems;
- To understand the digital payment systems

Course Outcomes (COs)

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

1. Recognize the impact of Information and Communication technologies, especially of the Internet in business operations
2. Recognize the fundamental principles e-Commerce
3. Understand the necessary infrastructure and functional components to develop Ecommerce systems
4. describe the key features of Internet, Intranets and Extranets and explain how they relate to each other.
5. Apply cryptographic technology and Web security protocols
6. Explain the added value, risks and barriers in the adoption of electronic fund transfer

Unit I- INTRODUCTION TO ELECTRONIC COMMERCE

What is E-Commerce (Introduction And Definition), Main activities E-Commerce, Goals of E-Commerce, Technical Components of ECommerce, Functions of E-Commerce, Advantages and disadvantages of E-Commerce, Scope of E-Commerce, Electronic Commerce Applications, 9 Electronic Commerce and Electronic Business(C2C)(C2G,G2G, B2G, B2P, B2A, P2P, B2A, C2A, B2B, B2C)

Unit II- THE INTERNET AND WWW

Evolution of Internet, Domain Names and Internet Organization (.edu, .com, .mil, .gov, .net etc.) , Types of Network, Internet Service Provider, World Wide Web, Internet & Extranet, Role of Internet in B2B Application, building own website, Cost, Time, Reach, Registering a Domain Name, Web promotion, Target email, Baner, Exchange, Shopping Bots

Unit III - INTERNET SECURITY

Secure Transaction, Computer Monitoring, Privacy on Internet, Corporate Email privacy, Computer Crime(Laws , Types of Crimes), Threats, Attack on Computer System, Software Packages for privacy, Hacking, Computer Virus(How it spreads, Virus problem, virus protection, Encryption and Decryption, Secret key Cryptography, DES, Public Key Encryption, RSA, Authorization and Authentication, Firewall, Digital Signature(How it Works)

Unit IV- ELECTRONIC DATA EXCHANGE

Introduction, Concepts of EDI and Limitation, Applications of EDI, Disadvantages of EDI, EDI model, Electronic Payment System: Introduction, Types of Electronic Payment System, Payment Types, Value Exchange System, Credit Card System, Electronic Fund Transfer, Paperless bill, Modern Payment Cash, Electronic Cash

Unit V- PLANNING FOR ELECTRONIC COMMERCE

Planning Electronic Commerce initiates, Linking objectives to business strategies, Measuring cost objectives, Comparing benefits to Costs, Strategies for developing electronic commerce web sites

INTERNET MARKETING

The PROS and CONS of online shopping, The cons of online shopping, Justify an Internet business, Internet marketing techniques, The E-cycle of Internet marketing, Personalization e-commerce.

SUGGESTED READINGS

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

WEBSITES

1. https://ecommerceguide.com/guides/what-is-ecommerce/
2. https://www.tutorialspoint.com/internet_technologies/internet_security_overview.htm
3. https://www.digitalvidya.com/blog/what-is-internet-marketing/

Semester – VI

19CSU602B COMPUTER GRAPHICS 4H – 4C

Instruction Hours / week: L: 4 T: 0 P: 0 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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. J.D.Foley, A.Van Dan, Feiner, Hughes Computer Graphics Principles & Practice. (1990). 2nd Edition. Publication Addison Wesley.
2. D.Hearn, Baker. (2008). Computer Graphics. Prentice Hall of India. 2008.
3. D.F.Rogers.(1997).Procedural Elements for Computer Graphics. McGraw Hill.
4. D.F.Rogers. (1989). Adams Mathematical Elements for Computer Graphics. McGraw Hill 2nd Edition.

WEBSITES

1. https://w3.cs.jmu.edu/bernstdh/web/common/references/graphics.php

2. www.cs.kent.edu/~farrell/cg02/reference/

3. www.cs.brandeis.edu/~cs155/Intro_6.pdf

Semester – VI

19CSU603A **ARTIFICIAL INTELLIGENCE** **3H – 3C**

Instruction Hours / week: L: 3 T: 0 P: 0 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

Course Objectives

- To gain a historical perspective of AI and its foundations.
- To become familiar with basic principles of AI toward problem solving, inference, perception, knowledge representation, and learning.
- To investigate applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
- To experience AI development tools such as an 'AI language', expert system shell, and/or data mining tool.
- To experiment with a machine learning model for simulation and analysis.
- To explore the current scope, potential, limitations, and implications of intelligent systems.

Course Outcomes (COs)

1. Demonstrate fundamental understanding of the history of artificial intelligence (AI) and its foundations.
2. Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning.
3. Demonstrate awareness and a fundamental understanding of various applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
4. Demonstrate proficiency developing applications in an 'AI language', expert system shell, or data mining tool.
5. Demonstrate proficiency in applying scientific method to models of machine learning.
6. Demonstrate an ability to share in discussions of AI, its current scope and limitations, and societal implications.

Unit I – INTRODUCTION

Introduction to Artificial Intelligence, Background and Applications, Turing Test and Rational Agent approaches to AI, Introduction to Intelligent Agents, their structure, behavior and environment.

Unit II - PROBLEM SOLVING AND SEARCHING TECHNIQUES

Problem Characteristics, Production Systems, Control Strategies, Breadth First Search, Depth First Search, Hill climbing and its Variations, Heuristics Search Techniques: Best First Search, A* algorithm, Constraint Satisfaction Problem, Means-End Analysis, Introduction to Game Playing, Min-Max and Alpha-Beta pruning algorithms.

Unit III - KNOWLEDGE REPRESENTATION

Introduction to First Order Predicate Logic, Resolution Principle, Unification, Semantic Nets, Conceptual Dependencies, Frames, and Scripts, Production Rules, Conceptual Graphs. Programming in Logic (PROLOG)

Unit IV- DEALING WITH UNCERTAINTY AND INCONSISTENCIES

Truth Maintenance System, Default Reasoning, Probabilistic Reasoning, Bayesian Probabilistic Inference, Possible World Representations.

Unit V- UNDERSTANDING NATURAL LANGUAGES

Parsing Techniques, Context-Free and Transformational Grammars, Recursive and Augmented Transition Nets.

SUGGESTED READINGS

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

WEBSITES

1. https://artint.info/html/ArtInt_350.html

2. https://www.cleverism.com/artificial-intelligence-complete-guide/

3. https://search.credoreference.com/content/topic/artificial_intelligence

Semester – VI

19CSU603B **SYSTEM PROGRAMMING** **3H – 3C**

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

End Semester Exam: 3 Hours

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, **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. Dhamdhere, D. M. (2011). Systems Programming. New Delhi: Tata McGraw Hill.
4. Leland Beck., & Manjula, D. (2008). System Software: An Introduction to System Programming (3rd 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.

WEBSITES

1. https://cs.gmu.edu/~setia/cs365-S02/assembler.pdf

2. https://www.geeksforgeeks.org/compiler-lexical-analysis/

3. https://www.javatpoint.com/parser

Semester – VI

19CSU611A **PHP PROGRAMMING - PRACTICAL** **4H – 2C**

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

End Semester Exam: 3 Hours

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 accept the number as an argument.

3. WAP to check whether the given number is prime or not.

4. Create a PHP page which accepts 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.

Semester – VI

19CSU611B UNIX / LINUX PROGRAMMING - PRACTICAL 4H – 2C

Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

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

Semester – VI

19CSU612A WEB AND E-COMMERCE TECHNOLOGIES – PRACTICAL 4H – 2C

Instruction Hours / week: L:0 T: 0 P: 4 Marks: Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

Course Objectives

- to describe Internet trading relationships including Business to Consumer, Business-to-Business, Intra-organizational.
- To analyse the impact of E-commerce on business models and strategy
- To describe the key features of Internet, Intranets and Extranets and explain how they relate to each other.
- To understand applied cryptographic technology and Web security protocols;
- To understand the necessary infrastructure and functional components to develop Ecommerce systems;
- To understand the digital payment systems

Course Outcomes (COs)

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

1. Recognize the impact of Information and Communication technologies, especially of the Internet in business operations
2. Recognize the fundamental principles e-Commerce
3. Understand the necessary infrastructure and functional components to develop Ecommerce systems
4. describe the key features of Internet, Intranets and Extranets and explain how they relate to each other.
5. Apply cryptographic technology and Web security protocols
6. Explain the added value, risks and barriers in the adoption of electronic fund transfer

Lab Practical: Web and E- Commerce Technologies LAB (based on the following topics): HyperText Markup Language (HTML): structural setup; page layout; text manipulation; special characters; images; links. Intermediate: image maps; tables; frames, forms; meta tags; web forms. Cascading Style Sheets (CSS): embedding/linking; HTML element selectors; classes; ID selectors, text manipulation; background; borders and spacing; layout; context selectors and grouping, pseudo-classes; pseudo-elements.

JavaScript : writing your first script; creating HTML tags; user input and output; loops and tables; payroll calculator, forms and text fields; validating an email address; radio buttons; check boxes; self-grading tests, image rollovers; slide shows; real-time clock; controllable clock; working with cookies.

Perl/CGI 10: sample Perl operations; random numbers; lists; dealing four poker hands; time manipulation; subroutines, hash tables; files; string matching, CGI; registration lists; surveys.

SQL and regular expressions: Regular expressions: basics; repeating; positioning. Beginner: select; where; order by; insert; update; delete, like; between; in; distinct; group by; aliases;

aggregate functions; create table; alter table; drop table., nested selects; SoundEx; join; deterministic functions; non-deterministic functions.

ASP structural setup: response.write; retrieving from forms; retrieving from querystring; variables; control constructs; subroutines and functions; session state; application variables; server variables; debugging, reading and writing cookies; server-side includes; response object methods; VBScript functions; error handling; debugging, browser details; CDONTS; files; output from a recordset; global.asa; setup instructions for using IIS and ASP. Flash 3 Create Flash movies of moving and interactive objects

Semester – VI

19CSU612B COMPUTER GRAPHICS – PRACTICAL 4H – 2C

Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

- 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.

Semester – VI

19CSU613A ARTIFICIAL INTELLIGENCE – PRACTICAL 3H – 1C

Instruction Hours / week: L: 0 T: 0 P: 3 **Marks:** Internal: **40** External: **60** Total: **100**
End Semester Exam: 3 Hours

Course Objectives

- To gain a historical perspective of AI and its foundations.
- To become familiar with basic principles of AI toward problem solving, inference, perception, knowledge representation, and learning.
- To investigate applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
- To experience AI development tools such as an 'AI language', expert system shell, and/or data mining tool.
- To experiment with a machine learning model for simulation and analysis.
- To explore the current scope, potential, limitations, and implications of intelligent systems.

Course Outcomes (COs)

1. Demonstrate fundamental understanding of the history of artificial intelligence (AI) and its foundations.
2. Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning.
3. Demonstrate awareness and a fundamental understanding of various applications of AI techniques in intelligent agents, expert systems, artificial neural networks and other machine learning models.
4. Demonstrate proficiency developing applications in an 'AI language', expert system shell, or data mining tool.
5. Demonstrate proficiency in applying scientific method to models of machine learning.
6. Demonstrate an ability to share in discussions of AI, its current scope and limitations, and societal implications.

List of programs

1. Write a prolog program to calculate the sum of two numbers.
2. Write a prolog program to find the maximum of two numbers.
3. Write a prolog program to calculate the factorial of a given number.
4. Write a prolog program to calculate the nth Fibonacci number.
5. Write a prolog program, insert_nth(item, n, into_list, result) that asserts that result is the list into_list with item inserted as the n'th element into every list at all levels.
6. Write a Prolog program to remove the Nth item from a list.
7. Write a Prolog program, remove-nth(Before, After) that asserts the After list is the Before list with the removal of every n'th item from every list at all levels.
8. Write a Prolog program to implement append for two lists.
9. Write a Prolog program to implement palindrome(List)
10. Write a Prolog program to implement max(X,Y,Max) so that Max is the greater of two numbers X and Y.

11. Write a Prolog program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List.
12. Write a Prolog program to implement sumlist(List,Sum) so that Sum is the sum of a given list of numbers List.
13. Write a Prolog program to implement two predicates evenlength(List) and oddlength(List) so that they are true if their argument is a list of even or odd length respectively.
14. Write a Prolog program to implement reverse(List,ReversedList) that reverses lists.
15. Write a Prolog program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List using cut predicate.
16. Write a Prolog program to implement GCD of two numbers.
17. Write a prolog program that implements Semantic Networks/Frame Structures.

Semester – VI

19CSU613B SYSTEM PROGRAMMING - PRACTICAL 3H – 1C

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

End Semester Exam: 3 Hours

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