

**RATHNAVEL SUBRAMANIAM COLLEGE OF ARTS & SCIENCE
(AUTONOMOUS) SULUR, COIMBATORE-641402**

**SCHOOL OF COMPUTER STUDIES
BACHELOR OF COMPUTER APPLICATION**



**Syllabus effective for the students admitted during
the Academic Year 2018-19 Batch & onwards**

(2018- 2021)



HOD



PRINCIPAL



COE

PROGRAMME OUTCOMES (POs):

PO1	Graduates can have strong fundamentals in their specific discipline along with DIGITAL STRATEGIC knowledge.
PO2	To increase student's ability to communicate effectively with the community /society in verbal /written courage for such as to give or receive clear instruction.
PO3	To enhance their ability to understand and identify the professional and ethical responsibilities.
PO4	To enrich their personality and character development

PROGRAMME SPECIFIC OUTCOMES: (PSOs)

Upon completion of Bachelor of Computer Applications Degree, STUDENTS are able to achieve the following outcomes.

PSO1	Perform the Job Roles such as UI/UX Developer, Software Developer and Back end Developer.
PSO2	Apply the skillsets of JavaScript, Web Designing, HTML, CSS, and AngularJS
PSO3	Interpret, Analyze and Design using Bootstrap, HTML5, CSS3 and JQuery
PSO4	Demonstrate the skillsets of Python/C/C++, Data Structure, Algorithms and SQL & NoSQL Database(MySQL, MongoDB).

GRADUATE ATTRIBUTES

- DISCIPLINE KNOWLEDGE
- PROBLEM ANALYSIS
- CRITICAL THINKING
- MODERN TOOLSUSAGE
- SOFT SKILLS
- SELF LEARNING
- LIFE LONG LEARNING
- INDIVIDUAL & TEAMWORK
- PROJECT MANAGEMENT & FINANCE

RATHNAVEL SUBRAMANIAM COLLEGE OF ARTS & SCIENCE

(Autonomous)

Affiliated to Bharathiar University, Coimbatore – 641 402

SCHEME OF EXAMINATION – CBCS PATTERN**PROGRAMME: BACHELOR OF COMPUTER APPLICATIONS**

(Effective from the academic year 2018-19)

Credits & Marks Distribution

Sl. No.	Course Type	Number of Courses	Credits	Marks	Total Credits
1	Multi Indian/ International Languages	2	4	200	8
2	Ability Enhancement Compulsory Courses – I & II : Group-I (English)	2	4	200	8
3	Ability Enhancement Compulsory Courses – II & IV: Group-II	2	1+3	200	4
4	Discipline Specific Courses	12	6	1200	72
5	Discipline Specific Elective Courses	4+1	6	500	30
6	Extra Disciplinary Course (EDC)	1	6	100	6
7	Skill Enhancement Courses	2 + 1	4	200	8
8	ALCTA- e Learning in MOOC platform	1	4*	Completion	4*
9	Non Credit Courses – Group I	2	-	Grade	-
10	Non Credit Courses – Group II	6	-	Pass	-
11	Non Credit Courses – Group III	4	-	Completed	-
Total				2600	136 + 4*

MULTI-INDIAN/ INTERNATIONAL LANGUAGES (MIL)								
Course	Course Name	L	T	P	CIA	SEE	Total	Credits
Two courses - Any one group								
Group I								
MIL-I	Tamil -I	6	-	-	25	75	100	4
MIL-II	Tamil- II	6	-	-	25	75	100	4
Group II								
MIL-I	Hindi- I	6	-	-	25	75	100	4
MIL-II	Hindi- II	6	-	-	25	75	100	4
Group III								
MIL-I	Malayalam- I	6	-	-	25	75	100	4
MIL-II	Malayalam- II	6			25	75	100	4
Group IV								
MIL-I	French-I	6	-	-	25	75	100	4
MIL-II	French-II	6	-	-	25	75	100	4
Group V								
MIL-I	Arabic- I	6	-	-	25	75	100	4
MIL-II	Arabic -II	6	-	-	25	75	100	4
Total							200	8

ABILITY ENHANCEMENT COMPULSORY COURSES (AECC) - GROUP I : (I & II SEMESTER)								
AECC - G1-I	English-I (Grammar and Usage)	6	-	-	25	75	100	4
AECC - G1-II	English-II (Communicative English)	6	-	-	25	75	100	4
Total							200	8

ABILITY ENHANCEMENT COMPULSORY COURSES (AECC) - GROUP II : (II & IV SEMESTER)

AECC -G2-I	Environmental Studies	1	-	-	100	-	100	1
AECC -G2-II	Aptitude	3	-	-	100	-	100	3
Total							200	4

DISCIPLINE SPECIFIC COURSES (DSC)

DSC - I	C Programming	4	-	4	25	75	100	6
					40	60		
DSC-II	Mathematics for Computer Science -I	5	1	-	25	75	100	6
DSC- III	Introduction to computing using Python	4	-	4	25	75	100	6
					40	60		
DSC- IV	Mathematics for Computer Science -II	5	1	-	25	75	100	6
DSC -V	Data Structures	4	-	4	25	75	100	6
					40	60		
DSC- VI	Operating Systems and Computer Networks	5	1	-	25	75	100	6
DSC -VII	Design and Analysis of Algorithms	4	-	4	25	75	100	6
					40	60		
DSC -VIII	Agile Software engineering	5	1	-	25	75	100	6
DSC -IX	Database Management System	4	-	4	25	75	100	6
					40	60		
DSC - X	JavaScript and JQuery	5	1	-	25	75	100	6
DSC - XI	NoSQL Database	4	-	4	25	75	100	6
					40	60		
DSC - XII	Responsive web design	5	1	-	25	75	100	6
TOTAL							1200	72

DISCIPLINE SPECIFIC ELECTIVE COURSES (DSE) I : (III SEMESTER)

Course	Course Name	L	T	P	CIA	SEE	Total	Credits
One course – From the group								
DSE-I	Foundations to User Experience	4	-	4	25	75	100	6
					40	60		
DSE-I	AngularJS	4	-	4	25	75	100	6
					40	60		
Total							100	6

DISCIPLINE SPECIFIC ELECTIVE COURSES (DSE) II : (IV SEMESTER)

Course	Course Name	L	T	P	CIA	SEE	Total	Credits
One course–From the group								
DSE- II	User Interface Design	4	-	4	25	75	100	6
					40	60		
DSE- II	PHP & MySQL	4	-	4	25	75	100	6
					40	60		
Total							100	6

DISCIPLINE SPECIFIC ELECTIVE COURSES (DSE) III : (V SEMESTER)

Course	Course Name	L	T	P	CIA	SEE	Total	Credits
One course –From the group								
DSE- III	Data analysis using Excel	4	-	4	25	75	100	6
					40	60		
DSE -III	Android programming I	4	-	4	25	75	100	6
					40	60		
DSE- III	Cyber security I	4	-	4	25	75	100	6
					40	60		
Total							100	6

DISCIPLINE SPECIFIC ELECTIVE COURSES (DSE) IV: (VI SEMESTER)

Course	Course Name	L	T	P	CIA	SEE	Total	Credits
One course –From the group								
DSE- IV	Data Visualization using Excel	4	-	4	25	75	100	6
					40	60		
DSE -IV	Android programming II	4	-	4	25	75	100	6
					40	60		
DSE- IV	Cyber security II	4	-	4	25	75	100	6
					40	60		
Total							100	6

DISCIPLINE SPECIFIC ELECTIVE COURSES (DSE) VI: (VI SEMESTER)								
DSE - V	Elective-V Project and Viva Voce	6	-	-	40	60	100	6
Total							100	6

DISCIPLINE SPECIFIC ELECTIVE COURSES (DSE): (V SEMESTER)								
EXTRA DISCIPLINARY COURSE : (EDC)								
Any one Course from the following								
EDC	Commercial Correspondence	5	1	-	25	75	100	6
EDC	Entrepreneurship	5	1	-	25	75	100	6
EDC	Project Management	5	1	-	25	75	100	6
EDC	Insurance and Risk Management	5	1	-	25	75	100	6
EDC	Social Media Marketing	5	1	-	25	75	100	6
EDC	E-Commerce	5	1	-	25	75	100	6
EDC	Indian Tax System	5	1	-	25	75	100	6
EDC	Digital marketing	5	1	-	25	75	100	6
EDC	Java Script and JQuery	5	1	-	25	75	100	6
EDC	Web Designing	5	1	-	25	75	100	6
EDC	Cyber Security	5	1	-	25	75	100	6
EDC	Hospitality Management	5	1	-	25	75	100	6
EDC	Fundamentals of Digital Computers	5	1	-	25	75	100	6
EDC	Computational Mathematics	5	1	-	25	75	100	6
EDC	Health Management	5	1	-	25	75	100	6
EDC	Forensic Science	5	1	-	25	75	100	6
EDC	Microbes – Health & Disease	5	1	-	25	75	100	6
EDC	Health & Life Style Disorders	5	1	-	25	75	100	6
Total							100	6

SKILL ENHANCEMENT COURSES : GROUP I (III & IV SEMESTER)								
SEC-G1- I	Communicative Skills -I	2	-	-	50	-	50	2
SEC-G1-II	Communicative Skills-II	2	-	-	50	-	50	2
Total							100	4

SKILL ENHANCEMENT COURSES : GROUP II (V SEMESTER)								
Course	Course Name	L	T	P	CIA	ESE	Total	Credits
Any one group								
Group A								
SEC- G2-A-I	Placement - College to Corporate I	2	-	-	50	-	50	2
SEC – G2-A-II	Placement - College to Corporate II	2	-	-	50	-	50	2
Group B								
SEC- G2-B	e-Learning in MOOC Platform	4	-	-	100	-	100	4
Total							100	4

NON CREDIT COURSE - GROUP I (III & IV SEMESTER)				
NCC-G1-I	Career Skills-I	RVS Training Academy		Grade
NCC-G1-II	Career Skills-II	RVS Training Academy		Grade

NON CREDIT COURSE - GROUP II (COP)									
Semester	Course Opted	Course Name	D	L	T	P	CIA	SEE	Marks
I	NCC-G2-I	2D Design Concepts Paper-I	3	1	-	-	25	75	100
II	NCC-G2-II	2D Design Concepts Practical-I	3	-	-	1	25	75	100
III	NCC-G2-III	3D Fundamentals Paper –II	3	1	-	-	25	75	100
IV	NCC-G2-IV	3D Fundamentals Practical- II	3	-	-	1	25	75	100
V	NCC-G2-V	3D Industry Workflow Paper – III	3	1	-	-	25	75	100
VI	NCC-G2-VI	3D Industry Workflow Practical – III	3	-	-	1	25	75	100
Total									600

NON CREDIT COURSE – GROUP III (I - IV SEMESTER)				
Any one Course				
NCC-G3	National Service Scheme		NSS	GOOD/SATISFACTORY
	National Cadet Corps		NCC	GOOD/SATISFACTORY
	Sports		Physical Education	GOOD/SATISFACTORY
	Literacy & Cultural Club		Language Department	GOOD/SATISFACTORY
	Youth Red Cross /Red Ribbon Club		YRC	GOOD/SATISFACTORY
	Fine Arts Club		Language Department	GOOD/SATISFACTORY

EXTRA OPTIONAL CREDIT COURSE (ALCTA) I – VI SEMESTER - Advanced Learners Course in Thrust Areas			
Any one Course with 4 extra credits			
ALCTA	e-Learning in MOOC Platform	4 CREDITS	Completion

SCHEME OF EXAMINATION
2018-2021

Semester	Course Opted	Course Name	D	L	T	P	CIA	SEE	Marks	Credits	
I	MIL-I	Tamil-I/ Hindi-I/Malayalam- I / French-I/Arabic-I	3	6	-	-	25	75	100	4	
	AECC-G1-I	English I- Grammar and Usage	3	6	-	-	25	75	100	4	
	DSC-I	C Programming	3 3	4	-	4	25 40	75 60	100	6	
	DSC-II	Mathematics for Computer Science –I	3	5	1	-	25	75	100	6	
	NCC-G3	NCC/NSS/ SPORTS/CULTURALS	-	1	-	-	-	-	-	-	
	LIB	Library	-	1	-	-	-	-	-	-	
	Total						28		400	20	
	II	MIL-II	Tamil-II/ Hindi-II/ Malayalam-II/French-II/Arabic-II	3	6	-	-	25	75	100	4
		AECC-G1-II	English II - Communicative English	3	6	-	-	25	75	100	4
DSC-III		Introduction to Computing using Python	3 3	4	-	4	25 40	75 60	100	6	
DSC-IV		Mathematics for Computer Science –II	3	5	1	-	25	75	100	6	
AECC - G2-I		Environmental Studies	3	1	-	-	100	-	100	1	
NCC-G3		NCC/NSS/ SPORTS/CULTURALS	-	1	-	-	-	-	-	-	
TOTAL						28		500	21		

Semester	Course Opted	Course Name	D	L	T	P	CIA	SEE	Marks	Credits
III	DSC - V	Data Structures	3	4	-	4	25	75	100	6
			3				40	60		
	DSC - VI	Operating Systems and Computer Networks	3	5	1	-	25	75	100	6
	DSE-I	Elective-I	3	4	-	4	25	75	100	6
			3				40	60		
	SEC-G1-I	Communicative Skills- I	3	2	-	-	50	-	50	2
	NCC-G1-I	Career Skills-I	3	2	-	-	Grade			
	NCC-G3	NCC/NSS/ SPORTS/CULTURALS	-	1	-	-	-	-	-	-
	LIB	Library	-	1	-	-	-	-	-	-
Total						28		350	20	
IV	DSC - VII	Design and analysis of Algorithms	3	4	-	4	25	75	100	6
			3				40	60		
	DSC-VIII	Agile Software engineering	3	5	1	-	25	75	100	6
	DSE-II	Elective-II	3	4	-	4	25	75	100	6
			3				40	60		
	SEC-G1-II	Communicative Skills -II	3	2	-	-	50	-	50	2
	NCC-G1-II	Career Skills-II	3	2	-	-	Grade			
	AECC-G2-II	Aptitude	3	3	-	-	100	-	100	3
	NCC-G3	NCC/NSS/ SPORTS/CULTURALS	-	1	-	-	-	-	-	-
Total						30		450	23	

Semester	Course Opted	Course Name	D	L	T	P	CIA	SEE	Marks	Credits	
V	DSC -IX	Database Management System	3	4	-	4	25	75	100	6	
			3				40	60			
	DSC - X	JavaScript and JQuery	3	5	1	-	25	75	100	6	
	DSE-III	Elective-III	3	4	-	4	25	75	100	6	
			3				40	60			
	EDC	EDC-Elective	3	5	1	-	25	75	100	6	
	Any One Group										
	Group A										
	SEC-G2-A-I	Placement - College to Corporate I	3	2	-	-	-	50	-	50	2
	SEC-G2-A-II	Placement - College to Corporate II		2	-	-	-	50	-	50	2
	Group B										
	SEC-G2-B	e-Learning in MOOC Platform	3	4	-	-	-	100	-	100	4
	NCC-G3	NSS/NCC/SPORTS/CULTURALS	-	-	-	-	-	GOOD/SATISFACTORY			
Total						32			500	28	
VI											
DSC - XI	NoSQL Database	3	4			4	25	75	100	6	
		3					40	60			
DSC -XII	Responsive web design	3	5	1	-	25	75	100	6		
DSE-IV	Elective-IV	3	4	-	4	25	75	100	6		
		3				40	60				
DSE -V	Elective -V - Project & Viva Voce	3	-	6	-	40	60	100	6		
ALCTA	e-Learning in MOOC Platform	-	-	-	-	-	-	-	-	4*	
Total						28			400	24	
Total									2600	140	

*- Extra Optional Course (ALCTA)

ABBREVIATIONS

MIL	- Multi Indian/ International Languages
AECC-G1	- Ability Enhancement Compulsory Courses – I & II: Group - I (English)
AECC-G2	- Ability Enhancement Compulsory Courses – II & II: Group - II
DSC	- Discipline Specific Courses
DSE	- Discipline Specific Elective Courses
EDC	- Extra Disciplinary Course
NCC	- Non Credit Course
SEC	- Skill Enhancement Courses (Group-I & II)
ALCTA	- Advanced Learners Course in Thrust Areas– e Learning in MOOC platform

DSE-I - Discipline Specific Elective Courses I: (III Semester)

1. Foundations to User Experience
2. Angular JS

DSE-II - Discipline Specific Elective Courses II: (IV Semester)

1. User Interface Design
2. PHP & MySQL

DSE-III - Discipline Specific Elective Courses III: (V Semester)

1. Data analysis using Excel
2. Android Programming I
3. Cyber Security I

DSE-IV- Discipline Specific Elective Courses IV: (VI Semester)

1. Data Visualization using Excel
2. Android Programming II
3. Cyber Security II

DSE -V- Discipline Specific Elective Courses V: (VI Semester)

1. Project & Viva Voce

NCC - I (Non – Credit course) Group – I

The assessment will be done by RVS Training Academy and grade will be given based on internal evaluation in the respective semester

NCC – III (Non – Credit Course) Group – III

The students shall complete the activities in the concerned semester and completion status will be mentioned in their fifth semester mark statement. However, completing the activities listed in Group – II is mandatory to complete their degrees.

SEMESTER-I

Course Title (T) : C PROGRAMMING	Course Code : 13A
Semester : I	Course Group : DSC-I
Teaching Scheme in Hrs (L:T:P) : 4:0:0	Credits : 4 Credits
Map Code : E(THEORY- PROGRAMMING)	Total Contact Hours : 60
CIA : 25 Marks	SEE : 75 Marks
Programme : BSc.CS /B.Sc IT/BCA	# - Semester End Exam

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Understand the Fundamentals of Computing, Programming and Debugging	PSO1 & PSO4	7	U/R
CO2	Define the Procedural Programming, Variables, Logical and Mathematical Operators	PSO1 & PSO4	7	U/AP
CO3	Determine the Control Structures, Conditionals and Loops.	PSO1 & PSO4	8	AP
CO4	Define the Functions and Error Handling	PSO1 & PSO4	8	U/AP
CO5	Explain the Data Structures, Strings and List	PSO1 & PSO4	10	U
CO6	Understand the File Input, Output, Dictionaries	PSO1 & PSO4	10	U/AP
CO7	Design programs that uses Object-Oriented Concepts	PSO1 & PSO4	10	AP

UNIT- I

LECTURE HOURS :14

Introduction: Computers and Programming : Moore's Law Computer organization machine languages, assembly languages, high level languages, interpreter Built for Performance, Standardization, The new C standard (computer programs, programmers, software, Moore's law logical units of a computer(table) machine, assembly and high level languages, interpreter portable, some popular performance oriented C Applications (table), hardware platforms) - Phase 1-6, standard input-output and error streams comments #include preprocessor directive blank lines and white space the main function an output statement escape sequences (typical C development environment(diagram) comments, document programs, multi line comments, %...% multi line comments C preprocessor, standard input output header(<stdio.h>) white space function, body, left and right brace action, character string or message or literal, argument, statement, statement terminator escape character, escape sequence, new line, some common escape sequences(table)) - variable and variable definitions identifiers and case sensitivity syntax error prompting messages the scanf function and formatted inputs assignment Statement(Definitions, variables, integer, Example code: addition program identifier, case sensitive placing variable definitions among executables statements cause syntax error prompt message using printf with new line scanf, format control string, %d conversion specifier, address operator, Enter key, Interactive computing assignment statement, operands) **Memory, arithmetic and decision making:** Introduction: memory concepts integer division and the remainder operator arithmetic expressions in straight line form(type, value, memory location showing the name and value of a variable, destructive, memory location after both variables are input, non destructive, memory location after a calculation arithmetic operators(table), integer division, remainder operator straight line form) - parentheses for grouping sub expressions rules of operator precedence sample algebraic and Cexpressions (Parentheses are used in the same manner in algebraic expressions four rules for operator precedence, associativity, precedence of arithmetic operators(table) expression calculates arithmetic mean(average), equation of a straight line, expression contains remainder(%)).Introduction: Decision Making (decisions, if statement, condition, equality and relational operators (table), Example code: using if statements, relational and equality operators , Precedence and associativity of the operators (table), C's keywords (table)) - **Structured program development :** Introduction: Algorithms and Pseudo code (Procedure: actions and order, Example: rise and shine algorithm, action statements) - Introduction : Control Structures

Flow charts (Sequential execution, transfer of control, go to statement, control structures: sequence, selection and repetition structure special purpose symbols: rectangles, diamonds, rounded rectangles and small circles, Flow charting C's sequence structure(diagram), small circle symbol(connector), decision symbol(diamond)) - Selection Statements in C Repetition Statements in C (single selection statement, double selection statement, multiple selection statement single entry / single exit control statements, control statement stacking) - **Assignment, Increment and Decrement Operators:** Several Assignment Operators Unary Increment and Decrement Operators (Additional assignment operator - Arithmetic assignment operators (table) increment operator, decrement operator , pre increment or pre decrement and post increment or post decrement, Increment and Decrement Operators (table), Example code: pre incrementing and post incrementing, Precedence and Associativity of the operators(table))

UNIT – II

LECTURE HOURS : 16

Control Structures: Counter controlled repetition Sentinel controlled repetition Counter Controlled Repetition requirements (definite repetition, control variable sentinel values are used to control repetition name, initial value, increment (or decrement), final value) - for Statement Header Components General Format of a for Statement Comma-Separated Lists of Expressions (Example code: Counter controlled repetition with the for statement for Statement Header Components (diagram) General format of for and while statements) - switch Statement Details switch Statement Flowchart Ignoring Newline, Tab and Blank Characters in Input Constant Integral Expressions Notes on Integral Types (controlling expressions-case labels Flowcharting a Switch multiple selection Statement with breaks skip Newline-Tab - Blank Characters test Constant Integral Expressions several data types to represent integers) - do...while Statement Flowchart break Statement, continue Statement (Example code: Using the do...while repetition statement, Flowcharting the do...while Statement Example code: Using the break statement in a for statement, Example code: Using the continue statement in a for statement.) – **Functions:** Introduction: Program Modules in C Introduction: Math Library Functions(Functions, C standard library, programmer defined functions, calling function or caller, called function, Hierarchical boss-function /worker function relationship(diagram) argument, Commonly used math library functions(table)) - Introduction: Functions main's return Type Function maximum (local variables, parameters, software reusability, abstraction Example code: Creating and using a

programmer, defined) - **Logical and Assignment operators:** Logical AND (&&) Operator Logical OR Operator logical negation (!) operator (Truth table for the logical AND (&&) operator Truth table for the logical OR operator Truth table for operator ! (logical negation)) - Introduction: Passing Arguments by Value and By Reference (function, format of function definition Example code: Finding the maximum of three integers pass-by-value, pass-by-reference, side effects) - **Storage Classes, Scope Rules:** Local Variables Static Storage Class Introduction: Scope rules (Storage class specifiers: auto, register, extern and static, storage duration, linkage, automatic storage duration, static storage duration, automatic variables extern and static, several types of identifiers with static storage duration scope of an identifiers, function scope).

UNIT – III

LECTURE HOURS :10

Introduction: Arrays: Defining Arrays Defining an Array and Using a Loop to Initialize the Array's elements (position number, zeroth element, 12-element array(diagram), subscript, name, value, Operator precedence and associativity(table) ,) - Initializing an Array in a Definition with an Initializer List Difference Between Passing an Entire Array and Passing an Array Element(Array initializers, Example code: Initializing the elements of an array with an initializer list %p conversion specifier, scalars, const, Example code: Array name is the same as the address of the array's first element, Example code: Passing arrays and individual array elements to functions) - **Multidimensional Arrays:** Two-Dimensional Array Manipulations(multi dimensional arrays, tables, double subscripted array, m by n array, Double-subscripted array with three rows and four columns (diagram), Example code: Initializing multidimensional arrays, Example code: Double-subscripted array manipulations.) – **Pointers:** Declaring Pointers Initializing and Assigning Values to Pointers Pointer Representation in Memory (indirection, Directly and indirectly referencing a variable(diagram) NULL is a symbolic constant address operator, Graphical representation of a pointer pointing to an integer variable in memory (diagram)) - The Indirection (*) Operator Demonstrating the & and * Operators Pass-By-Value Pass-By-Reference (Representation of y and yPtr in memory (diagram), Indirection Operator or dereferencing operator Example code: Using the & and * pointer operators, Operator precedence and associativity(table) Cube a variable using pass-by-value Cube a variable using pass-by-reference with a pointer argument Representation of y and yPtr in memory (diagram), Indirection Operator or dereferencing operator Example code: Using the & and * pointer

operators, Operator precedence and associativity(table) Cube a variable using pass-by-value
 Cube a variable using pass-by-reference with a pointer argument) - Determining the Sizes of
 the Standard Types, an Array and a Pointer Introduction: Pointer Expressions and Pointer
 Arithmetic (sizeof, Example code: Applying sizeof to an array name returns the number of
 bytes in the array, Example code: Using operator sizeof to determine standard data type sizes
 Array v and a pointer variable vPtr that points to v (diagram), The pointer vPtr after pointer
 arithmetic (diagram), pointer to void(void *)) - **Arrays of Pointers:** String Copying with
 Arrays and Pointers (offset, pointer/offset notation, pointer/subscript notation, pointer
 subscripting, Example code: Using subscripting and pointer notations with arrays, Example
 code: Copying a string using array notation and pointer notation) - Definition: Arrays of
 Pointers (Arrays of Pointers array of string or string array, Graphical representation of the
 suit array (diagram).)- Using Function Pointers to Create a Menu-Driven System (Pointers to
 a Function, passed to functions, returned from functions, stored in arrays, assigned to other
 function pointers, Example code: Demonstrating an array of pointers to functions)

UNIT – IV

LECTURE HOURS :10

Character and String Handling: Introduction: Fundamentals of Strings and Characters
 Functions isdigit, isalpha, isalnum and isxdigit Functions islower- isupper-tolower and
 toupper (Character constants, character set, special characters, string literals or string
 constant, null character, string is a pointer Character-handling library (<ctype.h>) functions
 (table), Example code: Using functions isdigit-isalpha- isalnum and isxdigit Example code:
 Using functions islower-isupper- tolower and toupper) - Functions isspace-iscntrl-ispunct-
 isprint and isgraph Functions fgets and putchar Function getchar (Example code: Using
 Functions isspace-iscntrl-ispunct-isprint and isgraph Standard input/output library character
 and string functions (table), Example code: Using Functions fgets and putchar Example
 code: Using Function getchar) - Functions strcpy and strncpy Functions strcat and strncat
 (String-manipulation functions of the string-handling library(table), Example code: Using
 functions strcpy and strncpy Example code: Using Functions strcat and strncat) -
 Introduction: Comparison Functions of the String-Handling Library (String-comparison
 functions of the string-handling library(table), Example code: Using functions strcmp and
 strncmp) – **Structures:** Self-Referential Structures Defining variables of Structure Types
 (derived data types, struct, structure tag, structure type and members, structemployee2- Self-
 Referential Structures struct card) - Structure Tag Names Operations that can be performed

on structures (Structure Tag Name is optional assigning structure variables to structure to structure variables of the same type, taking the address(&) of a structure variable, accessing the members of a structure variable and using the sizeofoperator to determine the size of a structure variable) - Introduction: Initializing Structures Introduction: Accessing Structure Members (struct card aCard = { "Three", "Hearts" }; structure member operator(.) and the structure pointer operator (->), Example code: Structure member operator and structure pointer operator) - Introduction: Using Structures with Functions (Structures may be passed to functions by passing individual structure members, by passing an entire structure or by passing a pointer to a structure.) - Introduction: typedef (typedefstruct card Card).

UNIT – V

LECTURE HOURS :10

Unions and Enumeration: Union Declarations Operations That Can Be Performed on Unions Initializing Unions in Declarations (union, share same space storage, union definition four operations can be performed on unions Union may be initialized with a value of the same type as the first union member) - Demonstrating Unions Introduction: Enumeration Constants (Example code: Displaying the value of a union in both member data types enumeration constants, Example code: Using an enumeration) - **Files and Streams:** Introduction: Files and Streams (end-of-file marker, stream, standard input-standard output-standard error-file descriptor, open file table, file control block (FCB), stdin-stdout-stderr, fgetc()-fputc()-fgets()- fputs(), fscanf()-fprintf(), fread()- fwrite().) - **Creating and Reading Sequential File :** Introduction: Creating a Sequential Access File (Example code: Creating a Sequential-Access File, End-of-file key combinations for various popular operating systems(table), Relationship between FILE pointers, FILE structures and FCBs(diagram), File opening modes(table)) - Resetting the File Position Pointer (Example code: Reading and printing a sequential file, file position pointer, file offset) - **Command Line Arguments:** Introduction (argc and argv, Example Code: Using Command Line Arguments).

TEXT BOOK:

T1. C - How to program, Paul and Harvey Deitel, 7th Edition, Pearson Education-Inc., 2013.

REFERENCES:

R1. The C Programming Language Edition: 2nd Prentice Hall Software Series Brian W. Kernighan ,DennisM.Ritchie(1988)

R2. Programming in C Edition:2nd Pearson Education Ashok N. Kamthane(2011)

R3. https://onlinecourses.nptel.ac.in/noc15_cs15/preview, Year 2015,

Course Instructor: 1. Dr.SatyadevNandakumar, Department of Computer Science and Engineering I.I.T. Kanpur and 2. Prof.AmeyKarkare, Department of Computer Science and Engineering, I.I.T. Kanpur

R4. <http://www.geeksforgeeks.org/c/>

Course Title : C PROGRAMMING (P)	Course Code : 13P
Semester : I	Course Group : DSC-I
Teaching Scheme in Hrs (L:T:P) : 0:0:4	Credits : 2 Credits
Map Code : G (PRACTICAL - PROGRAMMING)	Total Contact Hours : 60
CIA : 40 Marks	SEE : 60 Marks
Programme : BSc.CS /B.Sc IT/BCA	# - Semester End Exam

1. **Assignment / Lab Exercise(s): Program Development in C**
 - a. Body Mass Index Calculator
 - b. Car-Pool Savings Calculator
 - c. Credit Limit Calculator
 - d. Salary Calculator, Hollow Square of Asterisks
2. **Assignment / Lab Exercise(s): Program Control and Functions**
 - a. Local Variables || Static Storage Class || Introduction: Scope rules
 - b. De Morgan's Laws
 - c. Diamond-Printing Program
 - d. Parking Charges
 - e. Separating Digits, Greatest Common Divisor
3. **Assignment / Lab Exercise(s): Arrays and Pointer**
 - a. Duplicate Elimination
 - b. Calculating Circle Circumference Using Function Pointers
 - c. Calculating Circle Area Using Function Pointers
 - d. Sphere Volume Using Function Pointers
4. **Assignment / Lab Exercise(s): Character and String Handling – Structures**
 - a. Counting the Occurrences of a Character
 - b. Counting the Letters of the Alphabet in a String
 - c. Create a structure named Customer
 - d. Access a structure named Customer
5. **Assignment / Lab Exercise(s): Unions and File Handling**
 - a. Access a union named Customer

- b. Access a union named Customer
- c. Create a file named nameage.dat and add records sequentially to structure named Person,
- d. Print the records from file named nameage.dat

TEXT BOOK:

T1. C - How to program, Paul and Harvey Deitel, 7th Edition, Pearson Education-Inc., 2013.

REFERENCES:

R1.The C Programming Language || Edition: 2nd || Prentice Hall Software Series || Brian W. Kernighan and Dennis M. Ritchie (1988)

R2.Programming in C || Edition:2nd || Pearson Education || Ashok N. Kamthane(2011)

R3. https://onlinecourses.nptel.ac.in/noc15_cs15/preview, Year 2015, Course Instructor: 1. Dr.SatyadevNandakumar, Department of Computer Science and Engineering I.I.T. Kanpur and 2. Prof.AmeyKarkare, Department of Computer Science and Engineering I.I.T. Kanpur

R4. <http://www.geeksforgeeks.org/c/>

Course Title : MATHEMATICS FOR COMPUTER SCIENCE -I	Course Code : 13B
Semester : I	Course Group : DSC-II
Teaching Scheme in Hrs (L:T:P) : 5:1:0	Credits: 6 Credits
Map Code : F(PROBLEM – ANALYSIS)	Total Contact Hours : 90
CIA : 25 Marks	SEE : 75 Marks
Programme: BSc.CS /B.Sc IT/BCA	# - Semester End Exam

No	Course Outcome (Cos):	PSOs	Cl.Ses	CL
CO1	Define the types of number bases.	PSO4	18	REMEMBER
CO2	Recall to perform arithmetic operations in number bases.	PSO4	10	REMEMBER
CO3	Demonstrate the problems in modular arithmetic.	PSO4	8	UNDERSTAND
CO4	Explain the concepts of sequence and series.	PSO4	18	UNDERSTAND
CO5	Identify convergence/ divergence of sequence and series.	PSO4	18	APPLY
CO6	Construct the graphs from the function values.	PSO4	18	APPLY

UNIT : I**LECTURE HOURS : 18**

NUMBER BASES: Introduction to number bases.(Definition) -Place value for integers:binary to decimal & decimal to binary(Related problems)-Place value for fractional numbers:(Definition)binary to decimal & decimal to binary(problems)-Addition in binary(Rules and problems)-Subtraction in binary(Rules and problems)-Multiplication in binary(Rules and problems)-Place value for integers:(Definition) Octal to decimal & decimal to Octal(Related problems)Place value for fractional numbers:(Definition) Octal to decimal & decimal to Octal(Related Problems)-Place value for integers:(Definition) hexadecimal to decimal & decimal to hexadecimal(problems)-Placevalue for fractional

numbers:(Definition)hexadecimal to decimal & decimal to hexadecimal(Related problems)-
Arithmetic operations in Octal-Arithmetic operations in hexadecimal

UNIT:II**LECTURE HOURS : 18**

MODULAR ARITHMETIC: Introduction to modular arithmetic.(Definition) - Computing $n \bmod k$ (Rules and related problems)-Addition mod k (Definition)-Additive identity mod k (Rules and Problems)-Additive inverse mod k (Related and problems)-Multiplication mod k (Definition)-Multiplicative identity mod k (Rules and problems)-Multiplicative inverse mod k (Related problems)-Exponentiation mod k (Definition)-Mod, rem and division(Related Problems)-Encryption using modular arithmetic(Related Problems)

UNIT III**LECTURE HOURS : 18**

SEQUENCES AND SERIES : Introduction to Sequences(Definition)-Sequences of numbers(Related problems)-Defining sequences-Arithmetic progressions(Related Problems)-Geometric progressions(Rules and Problems)-Introduction to Series(Definition)-Finite sum of arithmetic sequences(Definition and Related problems)Finite sum of geometric sequences(Definition and Related problems)

UNIT IV**LECTURE HOURS : 18**

CONVERGENCE AND DIVERGENCE : Infinite sequences and sums (Definition)-Limit of a sequence(Related problems)-Convergence of a sequence-(Definition and Related problems)-Divergence of a sequence-(Related problems)-Limit of a series-Convergent series-(Related problems)-Divergent series(Related problems)--Criteria for identifying convergent/ divergent sequence (Related problems).

UNIT V**LECTURE HOURS : 18**

GRAPH SKETCHING : Introduction to graph sketching(Definition)- Cartesian coordinates(Subtopic)-Spiral(Related problems)-Introduction to functions and graphs(Definition)-Plotting graphs– straight lines(Related problems)--Plotting graphs – quadratics(Related problems)--Plotting graphs – cubics(Related problems)--Plotting graphs– higher order polynomials(Related problems)--Plotting graphs– reciprocal(Related problems)-Plotting graphs– rational functions(Related problems)--Plotting graphs-piecewise(Related problems)- Transformations of graphs

TEXT BOOK

T1. MATHEMATICS FOR COMPUTER SCIENCE- I : (COURSERA ONLINE COURSE)

REFERENCE BOOK :

R1.SEQUENCES AND SERIES by Arumugam, Isaac, Somasundaram

SEMESTER - II

Course Title : INTRODUCTION TO COMPUTING USING PYTHON(T)	Course Code : 23A
Semester : II	Course Group : DSC-III
Teaching Scheme in Hrs (L:T:P) : 4:0:0	Credits : 4
Map Code : E(THEORY - PROGRAMMING)	Total Contact Hours : 60
CIA : 25 Marks	SEE : 75 Marks
Programme : BSc.CS /B.Sc IT/BCA	# - Semester End Exam

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Understand the Fundamentals of Computing, Programming and Debugging	PSO1 & PSO4	7	U/R
CO2	Define the Procedural Programming, Variables, Logical and Mathematical Operators	PSO1 & PSO4	7	U/AP
CO3	Determine the Control Structures, Conditionals and Loops.	PSO1 & PSO4	8	AP
CO4	Define the Functions and Error Handling	PSO1 & PSO4	8	U/AP
CO5	Explain the Data Structures, Strings and List	PSO1 & PSO4	10	U
CO6	Understand the File Input Output, Dictionaries	PSO1 & PSO4	10	U/AP
CO7	Design programs that uses Object-Oriented Concepts	PSO1 & PSO4	10	AP

UNIT - I**LECTURE HOURS: 12**

Computing: What Is Computing? (Introduction to Programming, Programming Is Everywhere)-Programming Vocabulary (Programs and Code, Input and Output, Compiling and Executing)- Programming Languages, Console vs. GUI (The Console, GUIs)- Introduction to Python, Setting Up (Python: A High-Level Language, Python: An Interpreted Language, Files and the Command Line, Using an IDE, Web-Based IDEs, Interactive Mode). **Programming:** What Is Programming? & Writing Code: Lines (The Programming Flow, Chaining Together Instructions, The Print Statement, Work in Small Chunks)- Writing Code: Lines in Python & Running Code: Compiling vs. Executing (Your First Program: Hello, World, Printing Other Values, The Programming Flow, Compiling, Executing)- Executing Code in Python & Evaluating Results (Encountering Errors, “Compiling” Python, The Python Interactive Mode, Errors, Incorrect Results). **DEBUGGING:** What Is Debugging? - Debugging in the Programming Flow Types of Errors & Types of Errors in Python (Compilation Errors, Runtime Errors, NameError, TypeError, AttributeError, SyntaxError)- Basic Debugging & Basic Debugging in Python (Print Debugging, Scope Debugging, Rubber Duck Debugging, Print Debugging in Python, Scope Debugging in Python).

UNIT - II**LECTURE HOURS: 12**

Procedural Programming: What Is Procedural Programming? & Procedural Programming in Python (Functional Programming, Object-Oriented Programming, Event-Driven Programming, Hello, World, Data Types and Variables, Logical Operators, Mathematical Operators)- Comments and Documentation & Comments and Documentation in Python (Comments, Self-Documenting Code, In-Line Comments, Code Block Comments, Self-Documenting Code) **Variables:** What Is a Variable? & Variables in Python (Examples of Variables, Different Kinds of Variables, Python and Typing, Naming Rules and Conventions)- Assigning Variables & Assigning Variables in Python (Give Values to Variables, Assign Values before Using Variables, Giving Variables to Values (Wrong!))- Data Types & Data Types in Python (Basic Data Types, Importance of Data Types, Common Types, The type() Function, Mixing Types)- Type Conversions in Python & Reserved Keywords in Python (Converting to Strings, Converting from Strings, User Input, Python’s Reserved Words, Misusing Reserved Words, What about functions?) **Logical**

Operators: What Are Logical Operators? & Relational Operators (Mathematical Operators, Logical Operators, Numeric Comparisons, Non-Numeric Equality Comparisons, Set Operators)- Relational Operators in Python (Numeric Equality Comparisons, Numeric Value Comparisons, Non-Numeric Equality Comparisons, Set Operators, Operators in Action)- Boolean Operators & Boolean Operators in Python , And, Or, Not, Combining Boolean Operators, Simplifying Conditionals) **Mathematical Operators:** What Are Mathematical Operators? & Mathematical Operators in Python (The Assignment Operator, Mathematical Operators, Additional Operators, The Basic Mathematical Operators, Modulus, Other Operators)- Self-Assignment and Incrementing & Self-Assignment and Incrementing in Python (Self-Assignment, Incrementing, Self-Assignment, Self-Assignment Shortcuts, Incrementing and Loops)

UNIT - III

LECTURE HOURS: 12

Control Structures: What Are Control Structures? & The Control Structures (What Do Control Structures Do?, Conditionals, Loops, Function, Exception Handling)- Indentation and Control Structures in Python (Indentation and Conditionals, Nested Indentation)- Scope & Scope in Python (Examples of Scope, Control Structures and Scope, Simple Scope in Python, The Dangers of Scope in Python). **Conditionals:** What Are Conditionals? & Conditionals in Python (If-Then, If-Then-Else, If-Then-Else-If, Multiple Else-Ifs, Conditionals Recap, Common Errors)- Conditionals and Operators & Conditionals and Operators in Python (Relational and Mathematical Operators, Boolean Functions, Boolean Operators, Set Membership Operators). **Loops:** What Is a Loop? & Traditional For Loops in Python (For Loops, While Loops, For Loops with Known Ranges, For Loops with Unknown Ranges)- For-Each Loops in Python & While Loops in Python (For-Each and Lists, For-Each and Other Types, Simple While Loops, While Loops and Number Guessing, Infinite Loops). **FUNCTIONS:** What Is a Function? (Power of Functions, Function Terminology: Calls and Definitions, Parts of a Function Definition, Parts of a Function Call)-An Analogy for Functions (Setting Up the Analogy, The Function Definition, The Function Call, Bigger Functions)- Simple Functions in Python & Functions with Returns and Parameters in Python (The Function Definition, The Function Call, A Function with a Return, A Function with a Parameter, A Function with Multiple Parameters). **Error Handling:** What Is Exception Handling? & Try-Catch-Finally (“Catching” Errors, When to Catch Errors, The Try Block, The Catch Block, The Finally Block)- Try and Except in

Python (The Try Statement, Catching Any Error, Catching a Specific Error, Catching Multiple Specific Errors)- Else and Finally in Python (Else for Error Handling, Else and File Input, Finally, Finally and Uncaught Errors, Nested Try-Catch-Else-Finally)

UNIT - IV

LECTURE HOURS: 12

Data Structures: What Are Data Structures? (Advanced Data Types, Lists and List-Like Structures, Unit Outline)- Passing by Value vs. Passing by Reference & Passing by Value and Reference in (Passing by Value: An Analogy, Passing by Reference: An Analogy, Terminology: By Reference, Integers: By Value or by Reference?, Other Data Types: By Value or by Reference?, Variable Assignments)- Mutability in Python (Mutability vs. Passing by Reference, Reassigning Immutable Data Types, Immutable Data Types: Functions vs. Local Assignments, Printing Memory Addresses)- A Brief Introduction to Methods (Functions vs. Methods, Methods in Practice, Equivalent Syntax) **Strings & Lists:** What Are Strings? & Declaring Strings in Python (String and Alphabets, Unicode Characters, Special Characters, Three Ways to Declare Strings)- String Concatenation and Slicing in Python & String Searching in Python (String Concatenation, String Slicing: Individual Characters, String Slicing: Substrings, Negative Indices, The In Operator, The Find Method, Parameters of the Find Method)- What Are Lists? & Tuples in Python (Properties of Lists, List Synonyms, Declaring Tuples, Reading Tuples, Usefulness of Tuples, Nesting Tuples)**File Input and Output:** What Is File Input and Output? & Reading, Writing and Appending (Output Complements Input, File Types, Getting Started: Opening and Closing Files, Reading, Writing, Appending)- Writing Files in Python & Reading Files in Python (Simple File Writing, Writing Lists, Another Way to Output, Appending to Files, Simple File Reading, Loading into Lists, Save and Load Functions)-Files (Preparing to Save and Load, Global Variables, The Save Command, The Load Command).

UNIT - V

LECTURE HOURS: 12

Objects & Algorithms What Are Objects? & Objects and Instances in Python (What Are Objects?, Objects vs. Instances, Declaring a Class , Creating Instances, Objects vs. Dictionaries)-Encapsulating Methods in Classes & Encapsulating Methods in Python (Encapsulating Methods, Common Method Types: Constructors and Destructors, Common Method Types: Getters and Setters, Constructors in Python, Getters and Setters, Encapsulating Other Functions)-Advanced Topics in Classes in Python & Polymorphism

and Inheritance and Abstraction, Oh My! (Combining Classes, Instance Assignments, Instances as Arguments, Making Actual Copies, Abstraction, Polymorphism, Inheritance)- What Are Algorithms? & Complexity and Big O Notation (Famous Algorithms, Algorithms and Programming Languages, Complexity in Algorithms, Big O Notation, Common Big O Values)- Recursion & Sorting Algorithms (Simple Recursion: Factorial, Intermediate Recursion: The Fibonacci Series, Advanced Recursion: Directory Exploration, Bubble Sort).

Text Book:

T1. Introduction to Computing: Python, McGraw-Hill Education LLC, David Joyner, 1st Edition, 2016.

T2. Course Provider: www.edx.org, Course Name: Introduction-computing-using-python, Instructor Name:David Joyner, University Name: Georgia Tech University, Course Link: <https://www.edx.org/course/introduction-computing-using-python-gtx-cs1301x>

REFERENCE(S):

R1. LearnPython.org

R2. <https://www.codecademy.com/learn/python>

R3. <https://interactivepython.org/runestone/static/pythonds/index.html>

R4. How to Think Like a Computer Scientist: Learning with Python 3 by Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers, 3rd edition, 2012

R5. Introduction to Computer Science using Python, Wiley India Publication, Charles Dierbach, 2015.

R6. Beginning Python, Wiley India Publication, James Payne, 2011

Course Title : INTRODUCTION TO COMPUTING USING PYTHON(P)	Course Code : 23P
Semester : II	Course Group : DSC-III
Teaching Scheme in Hrs (L:T:P) : 0:0:4	Credits : 2 Credits
Map Code : G(PRACTICAL PROGRAMME)	Total Contact Hours : 60
CIA : 40 Marks	SEE : 60 Marks
Programme : BSc.CS /B.Sc IT/BCA	# - Semester End Exam

•

1. **Assignment / Lab Exercise(s):** Programming Basic

(i) Create four variables: Develop a Python Program that Creates four variables named my integer, my decimal, my string, and my Boolean. Assign to each variable a value of the corresponding type (an integer for my integer, a string of characters for my string, etc.). You can choose whatever values you want and just make sure the variable names and types are correct.

(ii) CovertintoStrings: Develop a Python Program to the following scenario. In the code below, four variables are created: an integer, a float, a date, and a boolean. Create four new variables: integer_as_string, float_as_string, date_as_string, and boolean_as_string. Convert the corresponding variables to strings. So, boolean_as_string should have the string version of the current value of new_boolean

2. **Assignment / Lab Exercise(s):** Procedural Programming

Freezing: Develop a Python Program to the following scenario. In the code below, there are two variables created named temperature and Celsius. Line 1: temperature = -3.7 Line 2: celsius = True Above are given two variables, temperature is a float that holds a temperature and celsius is a boolean that represents whether the temperature is in Celsius, if it's False, then the given temperature is actually in Fahrenheit. Add your code below that prints "Freezing" if the values above represent a freezing temperature, and "Not freezing" if they

don't. In Celsius, freezing is less than or equal to 0 degrees. In Fahrenheit, freezing is less than or equal to 32 degrees. [Hint: Add your code after line number 2

3. **Assignment / Lab Exercise(s): Control Structures**

HideandSeek: Develop a Python Program that creates a function called `hide_and_seek`. The function should have no parameters and return no value; instead, when called, it should just print the numbers from 1 through 10, follow by the text "Ready or not, here I come!". Each number and the message at the end should be on its own line. Then, call the function. There should be no print statements outside the function. Add the following lines of code at the end for testing your code. `hide_and_seek() #Invoking functio`

4. **Assignment / Lab Exercise(s): Data Structures**

StringLengthTuples: Develop a Python Program that creates a function called `string_length`. The function `string_length` should have on parameter, a string. It should return a 2-tuple: The first item in the 2-tuple should be the string itself, and the second item should be the length of the string as given by the `len()` function. Add the following lines of code at the end for testing your code. `print(string_length("Hello, world!")) print(string_length("CS1301")) print(string_length("Some people pronounce it 'toople'. Others pronounce it 'tuhple'. Either is correct.))` If your function works correctly, this will originally print: ('Hello, world!', 13) ('CS1301', 6) ('Some people pronounce it 'toople'. Others pronounce it 'tuhple'. Either is correct.', 83

5. **Assignment / Lab Exercise(s): Object Oriented Programming**

Creating Instance: Develop a Python Program that creates instances to the following class named Student 1| class Student: 2| def __init__(self): 3| self.studentName = "" 4| self.GPA = 0.0 5| self.creditHours = 0 6| self.enrolled = True 7| self.classes = [] Add your code after line number 7 so that it creates a new student with the following values and print each values: Student's Name: "George P. Burdell" Enrolled: True GPA: 3.9 Credit Hours: 1334 Classes: ["CS1301", "PHYS3001", "ISYE3029"]

Text Book:

- T1. Introduction to Computing: Python, McGraw-Hill Education LLC, David Joyner, 1st Edition, 2016.
- T2. Course Provider: www.edx.org, Course Name: Introduction-computing-using-python, Instructor Name: David Joyner, University Name: Georgia Tech University, Course Link: <https://www.edx.org/course/introduction-computing-using-python-gtx-cs1301x>

REFERENCE(S):

R1. LearnPython.org

R2. <https://www.codecademy.com/learn/python>

R3. <https://interactivepython.org/runestone/static/pythonds/index.html>

R4. How to Think Like a Computer Scientist: Learning with Python 3 by Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers, 3rd edition, 2012

R5. Introduction to Computer Science using Python, Wiley India Publication, Charles Dierbach, 2015.

Course Title : MATHEMATICS FOR COMPUTER SCIENCE-II	Course Code : 23B
Semester : II	Course Group : DSC-IV
Teaching Scheme in Hrs (L:T:P) : 5:1:0	Credits : 6 Credits
Map Code : F (PROBLEM ANALYSIS)	Total Contact Hours: 90
CIA : 25 Marks	SEE : 75 Marks
Programme: B.Sc CS/BCA /B.Sc IT	# - Semester End Exam

No	Course Outcome	PSOs	Cl.Ses	CL
CO1	To find the optimum layout of conductive tracks using very large scale integration(VLSI) design software.	PSO2	14	U
CO2	To gain the knowledge of network problems that for shipping the commodity from source to destination.	PSO2	15	U
CO3	To develop the emphasis in probability and involve the students for professional research.	PSO2	6	U
CO4	To explore the correlation between two quantities.	PSO2	21	U
CO5	To help the computers to manipulate photographs for enhancement , merging or comparison.	PSO2	16	U
CO6	To design data structures in the form of tree which in turn utilized vertices and edges.	PSO2	18	U

UNIT-I

LECTURE HOURS :18

LPP: **Linear programming** : Introduction –Basic Definitions, General Mathematical Formulation for Linear programming- General form of LPP, Standard form of LPP- (Related problems), Graphical method- Related problems, Simplex Algorithm-Related problems.

UNIT II

LECTURE HOURS :18

Sequencing : Introduction-Basic concepts, n jobs and 2 machine –Related problems ,n jobs and 3 machine –Related problems, n machine 4 jobs –Related problems, n jobs m machine –

Related problems. **Transportation Models:** NWC rule –Related problems, Matrix Minima Method-Related problems. Vogel Approximation Method –Related problems.

UNIT III

LECTURE HOURS :18

Theory of probability : Mathematical Probability –Related problems, Statistical probability-Related problems, Addition law –Related problems, Multiplication law-Related problems.**Correlation:** Types of Correlation –Related problem, Correlation coefficient-Related problem, Rank Correlation –Related problems.

UNIT IV

LECTURE HOURS :18

Regression :Regression coefficients- Related Problems, Regression equations - Related Problems. **Logarithms :**Introduction- the constant and natural logarithm-properties of logarithms-the change of base formula for logarithms (simple problems only). **Exponentials:** Introduction-Exponential expressions- Solving exponential equations using properties of exponents solving exponential operations with logarithms. (simple problems only).

UNIT V

LECTURE HOURS :18

Graph theory: Basic terminology-Definition, Paths-Cycle & Connectivity- Types & Examples, Subgraphs- Diagram representation, Types of graphs- Definition, Representation of graphs in computer memory- Matrix representation. **Trees:** Properties of trees, properties, Binary trees- Types & examples, Traversing Binary trees- Types&Examples, Computer Representation of general trees- Types & Examples Finite State Automaton

TEXT BOOK

T1. Resource Management Techniques by V.Sundaresan, K.S.Ganapathy Subramaniam, K.Ganesan,,A.R.Publications.

T2.Business Mathematics and Statistics by P.R.Vittal ,, Margam publications, Chennai.

REFERENCE BOOK

R1. Engineering mathematics by N.K.Venkatasubramaniam, K.A.Lakshminarayan, V.Sundaram, K.Balasubramaniam, .

R2. Discrete Mathematics, by J.K.Sharma,Second edition-2005, Macmillan India Ltd.

SEMESTER III

Course Title	: DATA STRUCTURES (T)	Course Code	: 33A
Semester	: III	Course Group	: DSC-V
Teaching Scheme in Hrs (L:T:P)	: 4:0:0	Credits	: 4
Map Code	: E(THEORY -PROGRAMMING)	Total Contact Hours	: 60
CIA	: 25 Marks	SEE #	: 75 Marks
Programme	: BSc.CS/BCA/B.Sc(IT)	# - Semester End Exam	

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Understand the basic concepts of Data Structures	PSO4	6	U
CO2	Understand the basic concepts of Recursion, Backtracking	PSO4	6	U
CO3	Implement Linked list using python	PSO4	12	AP
CO4	Demonstrate the stack ADT using linked list, array data structure and interpret the algebraic expressions using stack	PSO4	12	Ap
CO5	Demonstrate the queue ADT using linked list, array data structure	PSO4	12	Ap
CO6	Understand the operations using tree ADT and solve the problems using tree	PSO4	12	Ap

UNIT - I

LECTURE HOURS:12

Data Structures: Introduction(Variables, Datatypes, Data Structures)- Abstract Data Types (ADTs)(Abstract Data Types) **Algorithm:** Introduction(What is an Algorithm? Why the Analysis of Algorithms, Goal of the Analysis of Algorithms. How to compare Algorithms)- Analysis(Type of Analysis, Asymptotic Notation, Asymptotic Analysis) **Recursion:** Introduction(What is a Recursion, Why Recursion, Format of a Recursive function)- Recursion and memory(Visualization)- Recursion Vs Iteration(Example Algorithms of Recursion) **Backtracking:** Introduction(What is Backtracking? Example Algorithms of Backtracking) **Practical:** Towers of Hanoi Puzzle, factorial, fibonacci series using

recursion(Towers of Hanoi Puzzle, factorial, fibonacci series using recursion)- Sum of rows and columns and diagonals using 2D array(Sum of rows and columns and diagonals using 2D array)- Matrix Operations(Program to perform matrix operations)

UNIT - II**LECTURE HOURS:12**

Linked Lists: Introduction (What is a Linked List?) - Linked List ADTs (Operations and why linked list?) **Arrays:** Arrays Overview (Why constant time for accessing array elements? Advantages and Disadvantages of Arrays, Dynamic Arrays) - Linked List Overview (Advantages and Disadvantages of Linked Lists) **Singly Linked List:** Creation and Traversal (Creating and Traversing the Linked list) - Singly Linked list Insertion (Inserting a new node at the beginning, end and middle of the list) - Singly Linked list Deletion (Deleting node at the beginning, end and middle of the list) **Variations of Linked List:** Circular Linked List (Insertion and deletion of Circular linked list) - Doubly Linked List (Insertion and deletion of Doubly Linked list)

UNIT - III**LECTURE HOURS:12**

Stack: Introduction(What is a Stack? How stacks are used) - Stack ADT (Operations) **Applications:** Expressions (Infix, postfix, Prefix) - Evaluating Arithmetic expressions (Conversion of Infix to Postfix, Evaluation of a Postfix expression.) **Problems:** Evaluating Arithmetic expressions(Infix to postfix conversion, Postfix Evaluation) **Implementation:** Array Implementation (Array Implementation) - Linked List Implementation (Linked List Implementation) - Comparison of Implementations(Comparing Array Implementation and Linked List Implementation)

UNIT - IV**LECTURE HOURS:12**

Queues: Introduction (What is a Queue? How are queues used) - Queue ADT (Operations) **Queue Implementation:** Array Implementation (Array Implementation) - Linked List Implementation (Linked List Implementation) **Variations of Queue:** Circular Queue, Deque (Introduction and working of queue) - Problems on Queue (Insertion, Deletion of queue)

UNIT - V**LECTURE HOURS:12**

Trees: Introduction (What is a tree?) – Glossary (Glossary) **Binary Trees:** Types of Binary Tree (Strict, Full and Complete Binary tree) - Properties of Binary Trees (Properties of Binary Trees) - Operations on Binary tree (Insert, Delete, Search and traverse) **Types of**

Trees: Generic Trees (Representation of generic trees) - Threaded Binary Trees (Structure) -
Expression Trees (Expression Trees)

REFERENCES:

T1. NarasimhaKarumanchi: “ Data Structure and Algorithmic Thinking with Python”,
CareerMonk Publications,2016 (**Unit 1 – Chapter 1,2, Unit 2 – Chapter 3, Unit 3 –
Chapter 4,5, Unit 4 – Chapter 6, Unit 5 – Chapter 6,7**)

T2. G A V Pai : “Data Structures and Algorithms: Concepts, Techniques and Applications” ,
McGraw Hill Education (India) Pvt Ltd.

**T3. NPTEL Course (National Programme on Technology Enhanced Learning) Joined
initiative of the IITs and IISC. “Programming, Data Structures and Algorithms in
Python”, Dr. MadhavanMukund, Professor and Dean of Studies,
https://onlinecourses.nptel.ac.in/noc16_cs11**

Course Title : DATA STRUCTURES(P)	Course Code : 33P
Semester : III	Course Group : DSC-V
Teaching Scheme in Hrs (L:T:P) : 0:0:4	Credits : 2 Credits
Map Code : G(PRACTICAL-PROGRAMMING)	Total Contact Hours : 60
CIA : 40 Marks	SEE # : 60 Marks
Programme : BSc.CS/BCA/B.Sc(IT)	# - Semester End Exam

1. Assignment / Lab Exercise(s): Arrays & Recursion Concepts of Program Development in C

- a. Towers of Hanoi Puzzle,
- b. Factorial, Fibonacci series using recursion
- c. Sum of rows and columns and diagonals using 2D array
- d. Matrix Operations(Program to perform matrix operations)

2. Assignment / Lab Exercise(s): Linked List:

- a. Create a Singly Linked
- b. Perform the Insertion operation.
- c. Perform the deletion operations

3. Assignment / Lab Exercise(s): Array & Linked List Implementation of Stack

- a. Create a Stack and perform the push and pop operations using array
- b. Create a Stack and perform the push and pop operations using Linked list
- c. Conversion of Infix to Postfix (To convert an infix expression into postfix expression)

4. Assignment / Lab Exercise(s): Array & Linked List Implementation of Queue.

- a. Create a queue and perform the insert and delete operations using array
- b. Create a queue and perform the insert and delete operations using Linked list

5. Assignment / Lab Exercise(s): Trees

- a. Printing the level order data in reverse
- b. Maximum element in a binary tree

Course Title : OPERATING SYSTEMS & COMPUTER NETWORKS (T)	Course Code : 33B
Semester : III	Course Group : DSC-VI
Teaching Scheme in Hrs (L:T:P) : 5:1:0	Credits : 6 Credits
Map Code : D (THEORY-APPLICATION)	Total Contact Hours: 90
CIA : 25 Marks	SEE # : 75 Marks
Programme : BSc.CS/BCA/BSc IT	# - Semester End Exam

COURSE OUTCOMES

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Describe the components, uses of Operating system and types of operating systems based on processing method.	PSO1	18	U
CO2	Summarize memory management strategies, Compare and contrast virtual address and physical address	PSO1		U
CO3	Determine the Scheduling algorithm using Priority, Round robin techniques.	PSO1	18	Ap
CO4	Abstract the concept of Interrupt security	PSO1	18	U
CO5	Generalize the Basic Concepts of Networks, Transmission media,Adapters	PSO1	18	U
CO6	Clarify the Basic Concepts of Network devices.	PSO1	18	U

UNIT - 1

LECTURE HOURS:18

Introduction to Operating Systems:Computer System Organization-(Computer System Operation Storage structure,I/Ostructure)-Operating System Structure-Simple Structure, Layered Approach, Microkernel, Modules)-Computer System Architecture-Simple Structure, Layered Approach, Microkernel, ModulesOS Usage-(The Layers in Systems , Hard ware Abstraction, Resource Management)-PC Hardware-Memory Addresses , IO Addresses ,Memory mapped IO addresses General Purpose Registers)-8088, 80386 microprocessor ,Backward Compatibility.**Program to**

Process:Executing Programs-(process)-Process memory map-(Stack , Heap , Data , Text)-Beyond the programs memory map-(Kernel)-Communicating with OS-(System call, System call vs procedure call)-System call Interface-(File System call)-**Sharing the CPU:**Multiprogramming-(Idle CPU cycles , starvation)-Multitasking-(Time slice)-Multiprocessors-(Processor core , thread , Chip, Race condition)-Synchronization-(Lock, Scheduling)-**OS and Isolation:**OS Isolation-(Why it is needed? , How it is achieved)-OS and Security-(Access Control ,Security Assessment)-**Introduction to processes:**Virtual address map of a process-(Process page table)-Kernel, kernel and user space, contiguous mapping of kernel-(Where does kernel reside , Max limit, Max size , Kernel about a data process)-**Tutorial : Sharing the CPU:**Multiprogramming , Multitasking-(App1 , App2 , App3 wants to access the CPU .App5 is running in the CPU. When the CPU is Idle? How the performance is increased.)

UNIT - II

LECTURE HOURS:18

Process Concept and Operations: The process-(Process in memory)-Process state-(New ,Running , Waiting , Ready , Terminated)-Process Control Block-(Process state , Program counter , CPU registers, CPU scheduling information , Memory management information, Accounting information, I/O status information.)-Threads-(thread)-Process Creation-(A tree of process)-Process Termination-(Exit system call , wait system call)-**Memory Management Strategies:**Swapping-(Swapping of two processes using a disk as a backing store)-Contiguous Memory Allocation-(Memory mapping and protection ,Memory allocation , Fragmentation)-Paging-(Basic method, Hardware support , Protection ,Shared pages)-**Virtual Memory:**Demand paging-(Basic concepts , performance of Demand paging)-Memory Management Unit-(Virtual address , Physical address)-Working of Virtual memory-(Page table , trap , Example.)-Page Replacement-(FIFO replacement , Optimal page replacement ,LRU page replacement)-**PC Booting:**Powering Up : Reset , BIOS, MBR, Bootloader , OS-(Reset , BIOS, MBR, Bootloader, OS)-**Scheduling Algorithms:**FCFS, SJF ,Priority Scheduling-(waiting time, response time,turn around time)-Starvation , Round Robin Scheduling-(Starvation , Time quantum , Processor sharing)-Multilevel Queue Scheduling-(Multilevel Queue Scheduling)-Multilevel Feedback QueueScheduling-(Multilevel Feedback Queue)-**Tutorial: Memory Management Strategies:**Contiguous Memory Allocation-(The size of the RAM is 180k. Process 1 and process 4 occupy the RAM. Process 1 has the size 60K.Size of Process 4 is 110K. Process 1 Completes its execution the area in RAM is deallocated. New process process 6 Has the size 65K cannot start ? Why? How to avoid underutilization of RAM.).

UNIT - III**LECTURE HOURS:18**

Interrupt:Hardware Interrupt, Interrupt Handling-(Programmable Interrupt controller,What more happens when there is an interrupt)-Software Interrupts, CPU Context switching-(Explanation , Example,Contextswitch, Timer Interrupt ,Process context, Context switch overhead)-**Multiple-Processor Scheduling:**Approaches to Multiple-Processor Scheduling-(Symmetric multiprocessing,Asymmetricmultiprocessing)-Processor Affinity-(Soft affinity , Hard affinity)-Load Balancing-(Push migration , pull migration)-Approaches to Multiple-Processor Scheduling-(Multi threaded multi core systems)-**Classic problem of Synchronization:**The Dining Philosophers problem-(Deadlock state , No dead lock, Resource allocation graph, Bankers algorithm)-**Security:**Security Goals-(Secrecy , Integrity , Availability)-Access control systems-(Security Policy , Security Model , Security Mechanism)-**Networking Overview:**Introduction to Networking,Network Types-(LAN,WAN,MAN)-Networking Terms,NetworkingFacts,Practice Questions-(Transmission Media, Network interfaces, Protocols.)-**Networking Topologies:**Network,TopologyFacts,Practice Questions-(Bus, Ring, Star, Mesh, Terminators)-**The OSI Model:**The OSI Model,The OSI Model Facts-(Provides a common language and reference points for network professionals, Aids in troubleshooting, Provides modularity)-**Tutorial - Network Topologies:**Star Topology-(A book store uses networking in their entire process and they use a hub to connect multiple devices in their network. What physical topology is used in their process and how?).

UNIT- IV**LECTURE HOURS:18**

The OSI Model:The OSI Model Layers ,The OSI Model Communications-(HTTP, Telnet, FTP, TFTP, SNMP,Formats or "Presents" the data)-OSI Layers Facts,Practice Questions-(Application, Presentation, Session, Transport, Network, Data Link,Physical)-**Numbering Systems:**Numbering System, Numbering System Facts,Practice Questions-(What are the possible values in a binary number, Difference between a binary number and a hexadecimal numbering system, Binary (Base 2), Octal (Base 8), Hexadecimal (Base 16)-**Transmission Media:**Twisted Pair,Twisted Pair Facts , Coaxial cable , Fibre optic-(Why are wires twisted together in twisted pair cables, Difference in CAT3 and CAT5 cables, CAT 3, CAT 5E, CAT 6, CAT 6A , Coaxial cable , Fibre optic)-Connect to an Ethernet Network,Practice Questions-(Ethernet LAN using Twisted Pair cables.)-**Network Adapters:**Network Adapters-(Connect a media convertors, Whatdoes FF-FF-FF-FF-FF-FF indicates)-Network Adapter Facts-(A transceiver, Demodulation, Transceiver modular, Media convertor)-Select and install a network

adapter-(Install a NIC.)-Connect a media converter-(Port Tx, PortRx)-Practice Questions-(Practice Questions)-**Tutorial : Network Adapters:**Media converter-(Which device is suitable for the following scenario?You have a server that has a 100BaseFX network interface card that you need to connect to a switch. The switch only has 100BaseTX switch ports).

UNIT- V

LECTURE HOURS:18

Network Devices:-Network Devices-(At which OSI model layer do wireless access points operate, what type of device do you use to translate from one network architecture to another.)-Network Connection DeviceFacts-(Hub, Bridge, Switch, Wireless AP)-Hub, Bridge, Switch, Wireless AP-(Home-PC, Use AC-to-DC poweradapter.)-Practice Questions-(Practice Questions)-**Internetwork Devices:**InternetworkDevices,Internetwork Devices Facts-(Router, Firewall, Layer 3 switch)-Select a Router-(SOHO network & Internet, Select correct cables.)-Practice Questions-(Practice Questions)-**IP Addressing:**IPAddressing,IP Address Facts-(What is an Octet, Configure IPaddress, How does VLSM works, IP address is a 32-bit binary numbers)-Subnetting , Subnetting Facts-(Increase the number of devices, Decrease the number of devices, subnetting uses custom subnet masks)-IP Address Assignment-(Subnet, Host address, Default Gateway, TCP/IP settings.)-IP Addressing Facts-(DHCP, Static (manual) assignment.)-**Tutorial : IP Addressing:**Address classes-(Which two of the following IP addresses are Class B addresses? Point out the reason for identifying that as Class B addresses. 224.15.55.190.65.2.129.0.0.115.33.0.0).

References:

- R1. NPTEL (National Programme on TechnologyEnhanced Learning) is a joint initiative of Online Course Name - Introduction to Operating Systems Course Instructor - Chester Rebeiro—is an Assistant professor at IIT Madras. He completed his PhD from IIT Kharagpur and a post-doc from Columbia University. His research interests are in cryptography, system security, especially hardware and operating system security. Course url:https://onlinecourses.nptel.ac.in/noc16_cs1
- R2. <http://digiterati.com/> - Digiterati was started in 1991 as India's first IT Finishing School for Information Technology at Chennai. Digiterati is Chennai's largest corporate training centre with a unique distinction of training more than 1500 graduates on a single day simultaneously for India's top IT companies, like HCL TCS CTS Scope International Wipro Tech Mahindra Ford etc.

R3. Operating System Concepts Abraham Silberschatz , Peter Baer Galvin ,Greg Gagne , Eighth Edition , Wiley Publication.

R4. Operating Systems Internals And Design Principles “ WilliamStallings,Seventh Edition , Pearson Publication.

Course Title : FOUNDATIONS TO USER EXPERIENCE(T)	Course Code : 33E
Semester : III	Course Group : DSE -I
Teaching Scheme in Hrs (L:T:P) : 4:0:0	Credits : 4 Credits
Map Code : D (THEORY – APPLICATION)	Total Contact Hours : 60
CIA : 25 Marks	SEE # : 75 Marks
Programme : BCA	# - Semester End Exam

No.	Course Outcome	POs & PSOs	Cl. Ses	CL
CO1	Summarize the basics of user-experience and elements to achieve a good user interface design	PSO1	12	U
CO2	Compute technologies and tools for designing wireframes and prototypes	PSO1	12	AP
CO3	Use working knowledge of GIMP tool	PSO1		
CO4	Identify user requirements from user scenarios and Design Layouts	PSO1	12	AN
CO5	Discriminate user requirements from user scenarios and Design Layouts	PSO1	12	AN
CO6	Use to apply rules and laws that makes an UI an easy go for users	PSO1	12	AP

UNIT – I**LECTURE HOURS:12**

Network Signaling: Network Signalling|Transmission Systems (What is the purpose of Network signalling, Difference between baseband and broadband.) - Network Signalling Facts (Return-to-Zero, Non-Return-to-Zero, Manchester,De-Multiplexing.) - **Network Protocols:** TCP/IP Protocol Suite|Common Network Services (How does TCP differ from UDP, How does a protocol suite differ from a protocol.) - Common TCP/IP Protocols (Web Services, HTTP, HTTP over SSL, SSL, Security Protocols, TLS, File Transfer, POP3, IMAP, IMAP4.) – **Coaxial:** Coaxial|Coaxial Cable Facts (What is the function of the wire mesh in coaxial cables, which part of the cable is used to carry data, RG58, RG59, RG6, BNC.) - **Fiber Optic:** Fiber Optic|Fiber Optic Facts (Totally immune to EMI, Very expensive, Highly resistant to eavesdropping, Difficult to work with.) - **Wiring Implementation:** Twisted pair cable construction|Cable construction facts (Straight-through, Crossover, What is a patch panel used for.) - Wiring Distribution|Using

Punchdown blocks (Using Punchdown.) - Wiring Distribution Facts (Demarcation point, Main Distribution Frame, Demarc extension, Horizontal cross connect, vertical cross connect.) - **Troubleshooting Network Media:** Troubleshooting Copper Wires Issues||Copper Wiring Troubleshooting facts (EMI and RFI, Crosstalk, Near End Crosstalk, Far end Crosstalk.) - TroubleshootingFiber opting Wiring Issues||Fiber Optic Wiring Troubleshooting facts (Connectors, Polishing, Cabling, Media adapters, Attenuations.) - Troubleshooting tools||Troubleshooting tool facts (Loopback plug, Smart jack, cable tester, time-domain reflectometer.).

UNIT - II

LECTURE HOURS:12

Ethernet: Ethernet Architecture (What logical topologies are supported on an Ethernet network, what device is used to enable full duplex, what is the purpose of the backoff on Ethernet network.) - Ethernet Facts (Topology, Networking Devices, Transmission Media, Media Access Method.) - **Ethernet Specifications:** Ethernet Specifications (Reconnect to an Ethernet Network, Ethernet, 10BaseT, 10BaseFL, 100BaseTX.) - **Connecting Network Devices:** Connecting Devices||Device Connection Facts (Domain 3.0 Ethernet, When would you use a rollover cable?) - **Troubleshooting Physical Connectivity:** Troubleshooting the Fault Domain (Explore physical connectivity) - Fault Domain Troubleshooting Facts||Troubleshooting the Link Status (Bus, Star, Ring, Mesh) - Link Status Troubleshooting facts||Exploring Physical Connectivity (Unlit, Activity, Collision, Red/Amber, Solid Green.) .

UNIT – III

LECTURE HOURS:12

IP Addressing: Variable Length Subnetting facts, IP Addressing(Class A, Class B, Class C, classless addresses, Partial Subnetting, VLSM.) - IP Address Assignment (Subnet, Host address, Default Gateway, TCP/IP settings.) - IP Addressing Facts (DHCP, Static (manual) assignment.) - **Alternate IP Addressing:** APIPA (Configure an alternate IP configuration.) - Alternate IP Addressing facts (APIPA, Alternate IP Configuration.) - **DHCP Server Configuration:** Configuring a DHCP Server (Scope Name, Address Range, Subnet Mask) - Configure a DHCP Server (Exclusions and Delays, Lease Durations, Scope Option for the Router, WINS Server.) - Configuring a DHCP Options (Configure Subnet1 scope as follows, DNS Servers as 163.128.78.93, Domain Name as CorpNet.com) - Configure a DHCP Options (Configure a scope on the CorpDHCP12 server.) - Create DHCP Exclusions (Create a range to exclude addresses 192.168.0.1 to 192.168.0.6) - Create DHCP Client Reservations (Reservation Name, MAC address, IP Address.) - Configure Host Addressing (Configure Host Addressing) - Configure a

DHCP Client (Record the laptop's static IP, create an alternate TCP/IP connection.) - DHCP Configuration Facts (DHCP Discover, DHCP Offer, DHCP Ack, DHCP Request).

UNIT - IV

LECTURE HOURS:12

DHCP Relay: Configuring DHCP Relay (Difference between RFC 1542 compliant router and a DHCP relay agent, Configure a DHCP Relay Agent.) - DHCP Relay Facts (RFC 1542 Compliant Router, DHCP Relay Agent.) - Configure a DHCP Relay Agents (Add the DHCP Relay Agent Routing Protocol, Add vEthernet(External).) - Add a DHCP Server to an Another Subnet (ipconfig /all command, ipconfig /renew command.) - **Switch Access:** Device Access (Device Access) - Using the command Line Interface(CLI) (managed and unmanaged switches) - Device Connection facts (in-band management, out-of-band management) - Password Levels (privilege levels) - Configuring Line level passwords (console line password) - Configuring Enable Mode Passwords (enable password, enable secret) - Modify System Passwords (MF5, hashing algorithm, four VTY lines) - Configuring AAA Authentication (AAA authentication, TACACS server) - Switch Password Facts (console, VTY, EXEC mode)

UNIT - V

LECTURE HOURS:12

Switch IP Configuration: IP Address and Default Gateway Configuration (Why would you configure an IP address on a switch?) - Switch IP Configuration Facts (switch# config terminal, switch(config)#interface vlan1) - Configure Management VLAN Settings (Configure the VLAN1 interface, show run command) - Configure Switch IP Settings (start up-config file) - **Switch Interface Configuration:** Switch Operations (configure switch ports) - Switch Forwarding Facts (flooding the frame, unicast address, forwarding the frame, filtering the frame) - Switch Configuration Overview (switch configuration) - Switch Configuration Mode Facts (user exec, privileged exec, vlan database, global configuration) - Switch Configuration Command List (interface FastEthernet 0/14, interface GigabitEthernet 0/1) - Configure Switch Ports (Fa0/4 and Fa0/5, Fa0/8 and Fa0/23).

TEXT BOOKS:

T1. CCENT/CCNA ICND1 100-105, Official Cert Guide, Wendell Odom, CCIE NO.1624
Ciscopress.com

REFERENCES:

R1. Computer Networks - Andrew S Tanenbaum, Edition: 5th, Prentice Hall, 2011

Course Title : FOUNDATIONS TO USER EXPERIENCE (P)	Course Code : 33Q
Semester : III	Course Group : DSE-I
Teaching Scheme in Hrs (L:T:P) : 0:0:4	Credits : 2 Credits
Map Code : H (PRACTICAL EXPERIMENTS)	Total Contact Hours: 60
CIA : 40 Marks	SEE # : 60 Marks
Programme : BCA	# - Semester End Exam

LIST OF EXPERIMENTS

1. Peer-to-Peer Network
 - i. Task 1: Create a Peer-to-Peer Network.
 - ii. Task 2: Identify the Cables used in a Network.
 - iii. Task 3: Cable the Peer-to-peer Network.
 - iv. Task 4: Connect Your Workstations to the Classroom Lab Switch.
 - v. Task 5: Reflection

2. Cables and Connectors
 - i. Task 1: Connect the Devices in the Standard Lab Setup
 - ii. Task 2: View the Standard Lab Setup in the Physical Workspace

3. Ethernet
 - i. Task 1: IP Subnet Planning
 - ii. Task 2: Repair Problems with the Ethernet Switched LAN
 - iii. Task 3: Test the Network
 - iv. Task 4: To connect two different network by using default gateway

4. IP Addressing
 - i. Task 1: IP Subnet Planning
 - ii. Task 2: Finish Building the Network in Packet Tracer
 - iii. Task 3: Configure the Network
 - iv. Task 4: Test the Network

5. Configure DHCP
 - i. Configure DHCP pool in router R2 for Clients connected on its Gigabit Ethernet 0/1 interface by taking the following points into account:
 - ii. pool Name :pool1
 - iii. pool Length:192.168.88.0./28
 - iv. Default-Gateway:192.168.88.1
 - v. DNS-Server Address:4.2.2.2

6. Configure SSH
 - i. Configure SSH access to Router R1 using following parameters:
 - ii. Username: administrator
 - iii. Password: admin@456
 - iv. privilege level: 15
 - v. Domain: test.com

TEXT BOOKS:

T1. CCENT/CCNA ICND1 100-105, Official Cert Guide, Wendell Odom, CCIE NO.1624
Ciscopress.com

REFERENCE BOOK:

R1. Computer Networks - Andrew S Tanenbaum, Edition: 5th, Prentice Hall, (2011)

Course Title	: ANGULAR JS (T)	Course Code	: 33E
Semester	: III	Course Group	: DSE-I
Teaching Scheme in Hrs (L:T:P)	: 4:0:0	Credits	: 4 Credits
Map Code	: D(THEORY-APPLICATION)	Total Contact Hours:	60
CIA	: 25 Marks	SEE #	: 75Marks
Programme	: BCA	# -	Semester End Exam

No.	Course Outcome	POs & PSOs	Cl. Ses	BLOOM'S Taxonomy Level
CO1	Understand the design of single-page applications and how AngularJS facilitates their development.	PSO1	12	U
CO2	Separate the model, view and controller layers of application and implement using AngularJS.	PSO1	12	AP
CO3	Properly Angular Material is both a UI Component framework and a reference implementation of Google's Material Design Specification.	PSO1		
CO4	Identify user requirements from user scenarios and Design Layouts	PSO1	12	AN
CO5	Discriminate user requirements from user scenarios and Design Layouts	PSO1	12	AN
CO6	Modularize your code with the custom services and directives.	PSO1	12	AP

UNIT - I**LECTURE HOURS:12**

Introduction to AngularJS :Client-side JavaScript framework (Framework,User Interface,Expressive, Reusable and Maintainable Application Components). **Architectural concepts:** Model-View-Controller (MVC) (MVC,MVW,Language, Platform and Purpose of the

application). **What is a directive:** Reusable components (An Attribute, Element, Class and Comment). **Using AngularJS built-in directives:** The ngApp directive (The root of an AngularJS application to bootstrap the framework)- The ngController directive (The view and controller start to share the same scope and are ready to work together)- The ngBind directive(Span element and replaces the content of the element with the results of the provided expression)- The ngRepeat directive(Iterate over arrays and objects. The rows of a table, the elements of a list, and the options of select)- The ngModel directive(Attaches the element to a property in the scope, binding the view to the model)- The ngClick directive and other event Directives(Bind any custom behavior to the click event of the element)- The ngShow and ngHide directives(Changes the visibility of an element based on its display property) **Creating our own directives:** Template(The number of times the same snippet of the HTML code repeated over application code)- Template Url(The snippet to an isolated file and bind it using the templateUrl property)- Replace(Discard the original element, replacing it by the directive's template)- Restrict(The directives are restricted to be applied as attribute or Element). **Expressions-Basic usage with expressions:** Date(A date Value comes from the database or any other source in a raw and generic format)- Filter(Acting Over an array and applying any filtering criteria)- Lowercase (The content of the expression in lowercase)- Number(Format a string as a number)- Orderby(Order any array based on a predicate expression.String, array, function)- Uppercase(The content of the expression in uppercase). **Tutorial : PlugIns:** Configuring files(Configure the angular-min.js, angular-min.animate files etc).

UNIT - II

LECTURE HOURS:12

Form validation: Basic validation(The ngRequired directive, to intimate the validation process that the field is actually required)- Understanding the \$pristine and \$dirty properties(Pristine means purity that the field wasn't touched by anyone. After it's been touched for the first time, it becomes dirty.)- The \$error object(It accumulates the detailed list of everything that happens with the form.) **Dependency injection:** The level of Dependency(The object-oriented world, is known as coupling, and indicates the level of Dependency between the components.) **Creating services:** Creating services with the factory(Register the service in the application module that passes two parameters: the name of the service And the factory function.)- Creating services with the service(Uses a constructor function, which is equivalent to using the new operator.)- Creating services with the provider(The provider relies on the \$get function to expose its behavior) **Communicating with the backend:** HTTP, REST, and JSON(Interact with the backend was

Through http with the help of the get and post methods.) **Creating a single-page application:** Installing the module(The \$route service by mapping urls against controllers and views, and parameter passing)- Configuring the routes(Controller, templateUrl, resolve)- Rendering the content of each view(The \$route service and is responsible for rendering each template according to the routing mechanism)- Passing parameters(Inject the \$routeParams service, which will provide us with the parameters passed through the url) **The \$rootScope object- Scope Broadcasting:** Global Behavior(Injected inside any component such as controllers, directives, filters, and services)- \$broadcast(Communicate between components by the means of a scope) **Tutorial: JSON:** JSON Objects(Fetch array values by JSON Objects).

UNIT – III

LECTURE HOURS:12

Introduction: What is angularjs material?&Goals & Principles(UI component framework-Google's material design specification) **AngularJS Material Environmental setup:** Installing the angularjs material libraries(Configuring the library files for dependencies)- Build a material application (blank shell)(Configure the folders and files for application) **Introduction Material Design Layouts:** Layout and containers(Create modern, responsive layouts on top of CSS3 flexbox)- Layout and responsive break points(Associate breakpoints with mediaQuery definitions using breakpoint alias(es): Layout- xs,gt,sm,etc.,)- Layout API and Breakpoint overrides methods(Simple Layout markup convention. The alias is used as suffix extensions to the Layout API keyword) **Tutorial:Layout API:** Application Interface(Responsive Layouts)

UNIT : IV

LECTURE HOURS:12

Flex Directive:Responsive flex directives(The flex directive value is restricted to multiples of five, 33 or 66)- Additional flex values(Additional flex values provide to improve flexibility and to make the API easier to understand)- Ordering HTML elements(Order position within the layout container. Any integer value from -20 to 20 is accepted)- FLEX API and Breakpoint overrides methods(Device width when breakpoint overrides default)- Add offsets to the preceding HTML Elements(Flex-offset the margin-left is applied)- Set Child alignments within the layout container(The children aligned in the layout direction and perpendicular to the layout's direction) **Child Alignment:** Layout-Margin (Adds margin around each flex child)- Layout-Padding(Adds padding inside each flex child)- Layout-wrap(A non-trivial group of flex elements using layout-wrap)- Layout-fill(Forces the layout element to fill its parent container)-Show & Hide(The show

hide APIs to responsively show or hide elements) **Tutorial: Flex Directive:** Child alignment(Split the Layouts with Flex Directive)

UNIT : V

LECTURE HOURS:12

Directives: Autocomplete(To search for matches from local or remote data sources)- Bottom Sheet(Displayed by clicking one of the buttons below)- Button (Button directive with optional ink ripples)- Card (Directive is a container element used within `<md-content>` containers)- Checkbox(Directive is used like the normal angular checkbox)- Chips (Component used within `<md-chips>` and is responsible for rendering individual chips)- Content(Directive is a container element useful for scrollable content)- Dialog(The dialog's template must be inside this element)- Datepicker (The dialog's template must be inside this element)- Divider(Dividers group and separate content within lists and page layouts using strong visual and spatial distinctions)- FAB Toolbar(Directive is used to present a toolbar of elements for quick access to common actions when a floating action button is activated)- Input(Element as a child of an `<md-input-container>`. This allows you to build complex forms for data entry)- Icon(Directive makes it easier to use vector-based icons in app)- Input-Container(The parent of any input or textarea element)- List(Directive is a list container for 1..n `<md-list-item>` tags) -Nav Bar(Directive renders a list of material tabs that can be used for top-level page navigation)- Panel(Create dialogs, menus, and other overlays)- Radio Button(Directive is the child directive required to be used within `<md-radio-group>` elements)- Sidenav(Component that can be opened and closed programmatically)- Slider(Components allows the user to choose from a range of values)- Select(Component can be used within a `<md-input-container>` or as a stand alone component by using the `md-no-underline` class) -Switch(Enable or disable based on the expressions)- Subheader(Directive creates a sticky subheader for a section)- Toolbar(Place a toolbar in your app)- Toast(Toast can be dismissed with a swipe, a timer, or a button)- Tabs(Specify a tab with a label and optional view content)- Whiteframe(Apply an elevation shadow of element)- User-card(Card Avatar,Class for user image) **Tutorial Menus:** Menu Bar(Add Sub-Menus)

TEXT BOOKS:

T1.Angular JS Essential by Rodrigo Branas, Packt Publication.

REFERENCES

R1.<https://material.angularjs.org>

R2.<https://www.codeschool.com>

Course Title	: ANGULAR JS (P)	Course Code	: 33Q
Semester	: III	Course Group	: DSE-I
Teaching Scheme in Hrs (L:T:P)	: 0:0:4	Credits	: 2 Credits
Map Code	: M(PRACTICAL- APPLICATION)	Total Contact Hours:	: 60
CIA	: 40 Marks	SEE #	: 60Marks
Programme	: BCA #- Semester End Exam		

PROGRAM 1.Create a web page by using bulit-In directives.

PROGRAM 2.Develop an angularJS program for custom directive.

- i) Include template
- ii) templateUrl

PROGRAM 3.Create a login application by using angularjs material design.

- i) Add <md-input-container>
- ii) <md-button> to trigger the login

PROGRAM 4.Create a dashboard for a college and implement the following using UI component.

- i) Create controllers and directives
- ii) Use material design toolbar
- iii) Sidenav and content area
- iv) Responsive flex directives

PROGRAM 5.Develop a single page application by using the followings

- i) Create links by <md-list>
- ii) Map the links to specific page

TEXT BOOKS:

T1.Angular JS Essential by Rodrigo Branas, Packt Publication.

REFERENCES

R1.<https://material.angularjs.org>

R2.<https://www.codeschool.com>

SEMESTER IV

Course Title : DESIGN AND ANALYSIS OF ALGORITHMS (T)	Course Code : 43A
Semester : IV	Course Group : DSC-VII
Teaching Scheme in Hrs (L:T:P) : 4:0:0	Credits : 4 Credits
Map Code : E (THEORY-PROGRAMMING)	Total Contact Hours : 60
CIA : 25 Marks	SEE # : 75 Marks
Programme : BSc.CS /BCA/BSc.IT # - Semester End Exam	

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Understand the time complexity of various data structures	PSO4	12	U
CO2	Analyze the time complexity using Array and List	PSO4	12	AP
CO3	Describe and Analyze Heap Data Structure	PSO4	12	AP
CO4	Understand and Demonstrate the Searching and Sorting techniques	PSO4	12	AP
CO5	Describe and Demonstrate Binary Search Tree and Graphs	PSO4	12	AP
CO6	Understand Spanning Trees, Greedy Algorithms, Weighted Graphs and analyze the time complexity	PO1,PS O4	12	AP

UNIT - I

LECTURE HOURS : 12

Algorithm: Introduction (What is Algorithm?,How does it look like?, Example, Why Algorithms?) - Design of Algorithm (What is design of an algorithm?, Iterative and Recursive algorithms, The design steps, The design techniques, The role of Data structures in algorithm design.) **Asymptotic Complexity:** Analysis of Algorithms (What is Analysis?, Why Analysis?)- How to do Analysis? Efficiency and Running time an Algorithm (Definition, Big 'O' notation Measuring the running time of an algorithm, How to find an efficiency of an algorithm?)-Complexity (Definition, Space and Time complexity, Difference between performance and complexity, Complexity classes)-Worst case of an algorithm (What is worst case of an algorithm? How to determine complexity?)- Calculating complexity with iterative version (statements, If statements, single, double and triple loop (for) example)- Calculating complexity with Recursive version(While loop, Towers of Hanoi puzzle) **The two basic data structure:** Arrays and Lists(Introduction, storage of values, what is Array and List in storage point of

view?)- Array(Access an element with offset, Insert an element, the worst case input,Delete an element)- List(Access an element, Inserting an element using plumbing, Deleting an element)- Array vs. List(Difference between Array and List in complexity point of view) **Analysis of Stack and Queue:** Implementing stack using Array and List(Create, Inserting an element, Deleting an element, Traversing the datastructure)- Implementing queue using Array and List(Create, Inserting an element, Deleting an element, Traversing the datastructure) **Practical:** Calculate the running time of an Algorithm(Calculate the running time of an Algorithm)- Calculating Recursive and iterative time complexity of Towers of Hanoi(Calculating Recursive and iterative time complexity of Towers of Hanoi)- Analysis of Stack and Queue using list and arrays(Analysis of Stack and Queue using list and arrays)

UNIT - II

LECTURE HOURS : 12

Hash Table: Hash Table(Introduction, Storing values in Hash Table, Hashing)- Hash function(Choosing a good Hash function, Perfect Hash function, Load Factor)- Basic Operations and their Time complexity(Search, Insert, Delete)-Collision Resolution techniques(Linear Probing ,Separate Chaining, Applications of Hash)- Applications(Hash Applications) **Heap:** Heap Structure(Introduction, Heapify and it's analysis)- Types of heap(Max Heap, Min Heap)- Max Heap and it's Analysis(Find ,max, insert, delete)- Min Heap and it's Analysis(Find, min, insert, delete)- Other Heap operations(Merge, Meld, shift up and shift down)

UNIT - III

LECTURE HOURS : 12

Searching: Search Problem(Definition)- The Unsorted case(Linear Search – example)- The sorted case(Binary Search – example)- Comparison(Comparison) **Sorting:** Introduction(Definition, The need for sorting, Advantages)- Selection sort(sorting using swap)- Analysis(Recursive and Iterative versions)- Insertion Sort(The swap example)- Analysis(Recursive and Iterative versions)- Comparison of sequential sorting(Selection sort Vs. Insertion Sort Vs. Bubble Sort) **Divide and Conquer:** Introduction(The Divide and Conquer Strategy, Example, Advantages, Disadvantages)- Merge Sort(Example)- Analysis(Merge, Mergesort, Shortcomings)- Quick Sort(Introduction, Sort with Pivot, Tony Hoar's actual partitioning method)- Analysis(Randomization, Applications) **Stable Vs. Unstable sort:** Stable Sort(Definition, Example)- Unstable Sort(Definition, Example)- Which sorting is best?(Comparison)

UNIT - IV**LECTURE HOURS : 12**

Binary Search Tree: Operations(The structure of a BSTree, Find(v), Minimum, Maximum, Predecessor, Successor, insert, delete)- Analysis(Analysis of all operations) **Graphs:** Formal Representation(Directed Edge, Undirected Edge)- Working with Graphs Representation(Adjacency Matrix, Adjacency List) **Graph Explorations:** Breadth First Search(BFS)(Example, Algorithm)- Analysis(Adjacency Matrix, Adjacency List)- Depth First Search (DFS)(Example, Algorithm)- Analysis(Adjacency Matrix, Adjacency List)- Applications of BFS and DFS(Properties of DFS, Identifying connected components, Identifying cycles)

UNIT - V**LECTURE HOURS : 12**

DAG (Directed Acyclic Graph): Introduction(Directed Cycles, DAG definition, Features of DAG, Example) **Greedy Algorithm:** Greedy Strategy(Introduction, Comparison to the other techniques, Advantages) **Shortest Path:** Shortest Path(Weighted Graph, classification) - Single source shortest path (Dijkstra's Algorithm, Example, Analysis) **Spanning Trees:** Tree, Spanning Tree (Definition with example) - Spanning Tree with costs (Example) **Minimum Cost Spanning Tree:** Minimum Cost Spanning Trees (Prim's Algorithm)- Minimum Cost Spanning Trees(Kruskal's Algorithm)- Minimum Cost Spanning Trees(Analysis) **Practical:** Analysis of Greedy Algorithm (Analysis of Greedy Algorithm)- Analysis of Spanning Trees(Analysis of Spanning Trees)- Dijkstra's Algorithm(Analysis of Dijkstra's Algorithm)

REFERENCES:

- R1. Grokking Algorithms - An Illustrated guide for programmers and other curious people by Aditya Y. Bhargava, Manning Publications, 2016
- R2. Data Structures and Algorithms made easy by NarasimhaKarumanchi, Career Monk Publications, 2014
- R3. Computer Algorithms by Ellis Horowitz, SartajSahni, SanguthevarRajasekaran, Computer Science Press, 2011
- R4. NPTEL: <http://nptel.ac.in/courses/106106131>
- R5. <https://www.coursera.org/courses?languages=en&query=data+structures+and+algorithms>

Course Title : DESIGN AND ANALYSIS OF ALGORITHMS (P)	Course Code : 43P
Semester : IV	Course Group : DSC-VII
Teaching Scheme in Hrs (L:T:P) : 0:0:4	Credits : 2 Credits
Map Code : G (PRACTICAL-PROGRAMMING)	Total Contact Hours : 60
CIA : 40 Marks	SEE # : 60 Marks
Programme : BSc.CS /BCA/BSc.IT	# - Semester End Exam

1. Assignment / Lab Exercise(s): Programming Concepts of Array & Linked List

- a. Calculate the running time of an Algorithm
- b. Calculating Recursive and iterative time complexity of Towers of Hanoi
- c. Analysis of Stack and Queue using list and arrays

2. Assignment / Lab Exercise(s): Programming Concepts of Heap DS

- a. Calculating time complexity of Basic Operations in Heap(
- b. Max Heap Analysis & Min Heap Analysis

3. Assignment / Lab Exercise(s): Programming Concepts of Sort & Search

- a. Analysis of Selection sort
- b. Analysis of Insertion sort
- c. Analysis of Merge sort
- d. Analysis of Quick sort

4. Assignment / Lab Exercise(s): Programming Concepts of Binary Search Tree

- a. Analysis of Binary Search Tree
- b. Analysis of Depth First Search
- c. Analysis of Breadth First Search

5. Assignment / Lab Exercise(s): Programming Concepts of Greedy, Spanning Tree, Dijkstra's

- a. Analysis of Greedy Algorithm
- b. Analysis of Spanning Trees
- c. Dijkstra's Algorithm

Course Title : AGILE SOFTWARE ENGINEERING (T)	Course Code : 43B
Semester : IV	Course Group : DSC-VIII
Teaching Scheme in Hrs (L:T:P) : 5:1:0	Credits : 6
Map Code : D(THEORY-APPLICATION)	Total Contact Hours: 90
CIA: 25 Marks	SEE # : :75Marks
Programme: B.Sc CS / BCA/ B.Sc IT	#-Semester End Exam

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Generalize the principles and values of Agile	PSO3	18T	U
CO2	Determine the planning activities to minimize the probability of failure	PSO3	18T	AP
CO3	Discuss about frequent reviews of work products	PSO3	18T	U
CO4	Express the ground rules and internal process in order to strengthen team members	PSO3	18T	U
CO5	Estimate the project plan based on team learning	PSO3	18T	AP
CO6	Use appropriate techniques to solve risk management	PSO3		AP

UNIT-I**LECTURE HOURS: 18**

Introduction: Limitations of traditional project methodologies, How to overcome the Limitations of traditional projects? (What is expected, handle difficult situation, difficult to change the plan, avoid conflicts at later stages, produce incremental product at the end of every phase, transparent project communication.) What is agile?, Agile manifesto values and principles: Individuals and interactions over processes and tools, working software over comprehensive documentation,

customer collaboration over contract negotiation, responding to change over following a plan, the declaration of interdependence (Individuals and interactions, working software, customer collaboration, responding to change.) Agile methodologies, Agile principles and PMI- ACP domains: Waterfall VS Iterative VS agile (XP, DSDM, scrum, agile modeling, AUP, agile unified process, agile data method, Ess UP, FDD, open UP). Agile framework and terminology, Agile vs Plan-driven management, Agile Project life cycle: Iterations, incremental delivery, prioritization, Are all projects moving from traditional to agile methodology?, what kind of organization structure is best fit for agile project execution?, readiness assessment test for agile, introduction to agile planning , iteration(cycle) Execution (Time boxed cycle, release incremental product, prioritizes requirement, product vision, product backlog, project kickoff, iterative cycle, daily meeting, iterative cycle, iteration retrospective). SCRUM: Scrum roles, product owner, scrum master, team, scrum ceremonies, sprint planning, sprint review, sprint retrospective, daily scrum meeting/daily stand up, introspective, scrum artifacts, product backlog, sprint backlog, incremental product, burn-down charts, scrum challenges (Vision, project charter, product backlog, release backlog, sprint backlog). Communication in Agile Projects, Dimensions of Communication: General Communication Process, Levels of Listening (Sending messages, Receiving messages and Feedback Techniques, Active Listening). Barriers to Communications, Plan Communications: Communication within the Team, Information Radiator, Osmotic Communication, Cone of silence/caves and commons, Tacit knowledge, Daily stand- ups, Team Space, Communication technology (Types of communication, Iteration of work graph, Use cases or stories.) Communication Methods: Tips for building effective team communications, Rules for an effective team meeting, Collocation Vs Distributed team, Agile Tooling (Push communication, Pull communication, Interactive communication) Tutorial Hour: Agile Project Management: Product Vision, Product Backlog, Sprint review (Product owner leads, product backlog up-to-date).

UNIT-II

LECTURE HOURS : 18

Introduction, What is adaptive planning: Rolling Wave planning (Dynamic project environment the traditional predictive planning model to flexible adaptive planning.) Progressive Elaboration: Concept of user story, Format of user story, User story component, User stories example, 3 C's of user story(Reasons for adapting in agile projects, Understand the basic concepts of user story.) Iteration and Release planning , Daily planning: Iteration cycle, Estimation the tasks, Time

boxing (Iteration plan consists of Iteration backlogs, assumption, risks, dependencies and communication.) Agile project tracking: Information radiators, self-assessment, burn charts, cumulative flow diagram (Burn down chart, release burn down, cone of uncertainty theory, burn-up charts, combined charts, lead time, retrospectives.) Project cost management: Cost estimation, types of cost, problems in estimating accuracy cost, project funding requirements, control cost (Four common estimating styles, cost estimation based on accuracy level, budget , scurve.) Ideal Time, Affinity Estimation: Silent Relative sizing, Editing of wall, place items into relative sizing buckets, product owner challenge, get it into an electronic tool (Adjusting the plan until team ready to delivery stories.) Velocity/ Throughput/ Productivity, Lead time, Cycle time+: Improving velocity, Working cycle time (The term lead time and cycle time used in kanban method.) Tutorial Hour: Project Cost Management: Cost Estimation (Design to cost, Cost as the independent variable.)

UNIT-III

LECTURE HOURS : 18

Introduction , collecting and validating requirements of agile projects: Interviews, causes/ solutions, future visioning, mind map, Greenfield site, celebrity views, big picture, focus group, facilitated workshops, learning cycle (Interview with stakeholders, write the problem, ask participants to imagine, facilitate discussion.) Participatory decision model, product vision, product roadmap: Styles of group decision - making, command style, consultative style, consensus style (Nominal group technique, do what i say, all people support the decision finally, some members may still disagree with the decision) Backlog, story maps: Product backlog, product backlog refinement/ grooming, release backlog, sprint backlog (Cumulative of all stories which are yet to be implemented, formal agile ceremony, visibility to all stake holders, list of work the team will work.) Frequent Verification and Validation, Test Driven Development: Continuous Integration (The time and effort required each cycle is minimized.) Basic Concepts of Configuration Management, Definition of Done, Incremental Delivery: Testing, Verification testing, Validation testing, Exploratory testing, Usability testing, Acceptance Criteria of the customer for Incremental Delivery as Deliverable (Iterative cycle with each delivering part of the required functionality and highest priority.) Tutorial Hour: Configuration Management: Verification testing (Validation basic phases.)

UNIT-IV**LECTURE HOURS: 18**

Feedback Techniques for Product, Continuous Improvement Process: Process Improvement Strategy, Continuous Improvement Model, PDCA Cycle, Continuous Quality Improvement, Scorecard for Improvement (Improvement happens at different levels and at different stages of the program.) Five Whys, Cause and effect/ Fishbone Diagrams: Identify the problem, Work out the Major Factors Involved, Identify Possible Causes, Analyse your Diagram, Pareto Charts, SWOT Analysis (Cause and effect determine the root cause problems.) Emotional Intelligence Quotient, Collaboration: Team Collaboration and Team Commitment, Co- location and Distributed Team (Resolving issues and coordinating changes to enhance project performance) Negotiations: Negotiation Mode, Steps for Negotiation (Determine, Identify and strengths and weakness for issues.) Brainstorming Techniques, Building Empowered Team: Defining team, Types of team, agile team stages (Mentoring transfer agile knowledge and experience to the team.) Team Motivation and Knowledge Sharing, Leadership Styles, Problem-Solving Steps: Team Empowerment, Team Motivation (Approach, Team Communication to avoid problems.) Tutorial Hour: Problem Solving: Problem solving rubric (Technical investing to find answer to problem.)

UNIT-V**LECTURE HOURS: 18**

Definition of value, ROI, NPV, IRR: Minimal Process and JIT Process, Payback Period, ROI, NPV, IRR (Beliefs aims and assumptions that undergird the enterprise and guide management.) Value Driven Delivery: Define Positive Value, Define work and features in terms of value, Value Agreements (Value, action, functionality will be fixed in the sprint.) Elicit Non-functional requirements: Value based analysis (Develop Incremental value, shared value model.) Approaching Risk in Agile Projects, Identify Risks, Risk Adjusted Backlog, Risk Burn Down Graphs: Risk Categories (Business, Technical and logistical risk.) Perform Qualitative Risk Analysis: Risk Probability and Impact Assessment, Probability and Impact Matrix, Risk Data Quality Assessment, Risk Urgency Assessment, Risk Categorization (Time based, Symptoms and Warning signs of risk, risk rating.) Tutorial Hour: Risk based spike: Decision Tree Analysis (Expected monetary value analysis.)

Text book:

T1. PMI Agile Certified Practitioner- Excel with Ease, Third edition, S.Chandramouli, Saikat Dutt, 2017, Pearson India education services Pvt.Ltd

Course Title : USER INTERFACE DESIGN (T)	Course Code : 43E
SEMESTER : IV	Course Group : DSE -II
Teaching Scheme in Hrs (L:T:P) : 4:0:0	Credits : 4 Credits
Map Code : D (THEORY – APPLICATION)	Total Contact Hours : 60
CIA : 25 Marks	SEE # : 75 Marks
Programme : BCA	# - Semester End Exam

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Recall all the aspects of User Experience design concepts to develop an intuitive prototype of the Application.	PSO4	12	U
CO2	Create Layouts in HTML provides a basic understanding of the structure of the website.	PSO4	12	U
CO3	Construct Layouts in HTML 5 provides a basic perceptive of the composition of the website.			
CO4	Enhancements in Looks by Colours, fonts, alignment and texture are achieved through CSS.	PSO4	12	U
CO5	Meet all the additional Functionalities (pop-ups and forms) and Features through JQuery	PSO4	12	Ap
CO6	Provides overall understanding of transforming websites for multiple platforms (mobile and tablets)	PSO4	12	Ap

UNIT - I**LECTURE HOURS: 12****USER EXPERIENCE PRINCIPLES -USABILITY:** Usability and the User Experience (Explanation)

- What Is Usability? (Usability) - The components of usability (Learnability, Efficiency, Memorability, Errors, Satisfaction) - **SKETCHING LAYOUTS (MOBILE APPS AND WEBSITES)** : Mobile layouts, Website layouts (Classifications based on the genre) - **TYPOGRAPHY** : Fonts and sizes (Fonts, Arrangements, Spacing) - **COLOUR THEORY** : Colour Palates (digital and print) - Content Visualization (Explanation) - **GIMP** : Importing files and framing layouts (Examples) - Export File Formats (File formats).

UNIT - II**LECTURE HOURS : 12**

HTML INTRODUCTION : Basics (Introduction to mobile application) - Elements (HTML elements & attributes) - Attributes (Explanation) - Headings (Explanation) - Paragraphs (Explanation) - Formatting Links (Explanation) - Images (Explanation) - Tables (Explanation) - Lists (Explanation) - Layout (Explanation) - Forms (HTML Form elements) - Iframes (HTML iframes) - **HTML5** : Introduction (HTML5 code structure, new semantic elements) - New Elements - Semantic - **HTML5 FORMS** : Input Types - Form - Elements – Form attributes (HTML Form attributes) - **HTML5 MEDIA** : Media - Video - Audio - (HTML multimedia) - Plug-in- YouTube.

UNIT - III**LECTURE HOURS : 12**

CSS INTRODUCTION : CSS Introduction (What is CSS?) - Syntax (Selectors) - Id & Class (Types of CSS.) - **HOW TO** : Backgrounds (Syntax and Explanation) - Text (Syntax and Explanation) - Fonts (Syntax and Explanation) - Links (Syntax and Explanation) - Lists (Syntax and Explanation) - Tables (Syntax and Explanation) - Box (Syntax and Explanation) - Model (Syntax and Explanation) - Border (Syntax and Explanation) - Outline (Syntax and Explanation) - Margin (Syntax and Explanation) - Padding (Syntax and Explanation) - **CSS ADVANCED** : Grouping / Nesting (Syntax and Explanation) - Dimension (Syntax and Explanation) - Display (Syntax and Explanation) - Positioning (Syntax and Explanation) - Floating (Syntax and Explanation) - Align (Syntax and Explanation) - Navigation Bar (Syntax and Explanation) - **IMAGE GALLEY** : Image Opacity (Explanation) - **CSS3 INTRODUCTION** : Borders (Syntax and Explanation) - Backgrounds (Syntax and Explanation) - Multiple Columns (Syntax and Explanation) - **USER INTERFACE** : Browser Support (Explanation) - Selectors (Syntax and Explanation) - Fonts face (Syntax and Explanation) - Units (Syntax and Explanation) - Colors (Syntax and Explanation).

UNIT - IV**LECTURE HOURS : 12**

JQUERY HOME : Introduction (Explanation) - Install - Syntax (Explanation) - Selectors - Events (Explanation) - JQuery Effects Hide/Show (Explanation) - Fade - Slide - Animate - Stop() - Callback – Chaining (Explanation) - **JQUERY HTML** : Get -Set (Explanation) - Add - Remove - CSS Classes - css() (Explanation) - **AJAX** : AJAX Introduction (Explanation) - Load - Get / Post - - noConflict(). (Explanation) - **JQUERY MOBILE INTRODUCTION** : Install - Pages - transitions (Explanation) - Buttons - Icons - Popups (Explanation) - Toolbars - Navbars - Collapsibles - Tables – Grids (Explanation) - **JQUERY MOBILE LISTS** : List Views - List Content- (Explanation) - Filter Items

(Explanation) - **JQUERY MOBILE FORMS** : Form Baisc - Form Inputs (Explanation) - Form Select Form - Sliders (Explanation) - Themes (Explanation) - **JQUERY MOBILE EVENTS** : Events (Explanation) - Touch (Explanation) - Scroll – Orientation (Explanation).

UNIT - V

LECTURE HOURS : 12

BOOTSTRAP : Introduction (Explanation) - Quick start (Explanation) - CS, JS Config (Explanation) - HTML5 doctype (Explanation) - Responsive meta tag (Explanation) - BootstrapCDN (Explanation) - **LAYOUTS** : Overview (Explanation) - Containers (Explanation) - Responsive breakpoints (Explanation) - Grid system (Explanation) - **ALIGNMENT** : Order (Explanation) - Typography (Explanation) - Code (Explanation) - Images (Explanation) - Tables (Explanation) - Alerts (Explanation) - **BREADCRUMB** : Buttons (Explanation) - Button group (Explanation) - Cards (Explanation) - Carousel (Explanation) - Collapse (Explanation) - Dropdowns (Explanation) - Forms (Explanation) - Input group (Explanation) - **JUMBOTRON** : List group (Explanation) - Modal (Explanation) - Navs (Explanation) - Navbar (Explanation) - Icons (Explanation) - Pagination (Explanation).

References:

- 1.**HTML & HTML5** for Android Apps Development VOL-1.
- 2.**CSS & CSS 3** A user friendly reference for guide VOL-1.

Course Title : USER INTERFACE DESIGN (P)	Course Code : 43Q
Semester : IV	Course Group : DSE-II
Teaching Scheme in Hrs (L:T:P) : 0:0:4	Credits : 2 Credits
Map Code : H (PRACTICAL EXPERIMENTS)	Total Contact Hours : 60
CIA : 40 Marks	SEE # : 60 Marks
Programme : BCA	# - Semester End Exam

Practical Lab Programs :**60 Hours**

Program-I : Create a visualization for the given content

Program-II : Create web banner using GIMP

Program-III : Create a basic html page

Program-IV : HTML webpage with advanced tags

Program-V : Style the web page with given Colour palate

Program-VI : Create a image slider for the news scrolling

Program-VII : Work with on hold and hide controls with jQuery

Program-VIII : Customize your web page responsive using bootstrap

Program-IX : Make you web page responsive using bootstrap

Program-X : Write all possible media-queries for multi-platform using bootstrap

References:**R1. HTML & HTML5** for Android Apps Development VOL-1.**R2. CSS & CSS 3** A user friendly reference for guide VOL-1.

Course Title : PHP and MySQL (T)	Course Code : 43E
Semester : IV	Course Group : DSE-II
Teaching Scheme in Hrs (L:T:P) : 4:0:0	Credits : 4
Map Code : D (THEORY-	Total Contact Hours : 60 Hrs
CIA : 25 Marks	SEE # : 75 Marks
Programme : B.C.A	# - Semester End

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Understand Server Side web Development	PSO1	12	U
CO2	Learn to code a PHP application	PSO1	12	AP
CO3	Acquire knowledge on Relational database and Mysql database	PSO1	12	AP
CO4	Understand MVC patterns to organize the application	PSO1	7	U
CO5	Acquire knowledge to test and debug the PHPApplications.	PSO1	5	U
CO6	Learn to work with form data	PSO1	12	AP

UNIT - I**LECTURE HOURS : 12**

Introduction to web development with PHP : The architecture of a web application (How a client-server architecture works, How static web pages are processed, How dynamic web pages are processed, A survey of web application software. Highlights in the history of PHP, Highlights in the history of MySQL) - The Product Discount application (The user interface, The HTML file, The CSS file, The PHP file) - How to edit and test a PHP application (How to edit a PHP page with a text editor, How to start and stop Apache and MySQL on your own computer, How to deploy a PHP application, How to run a PHP application, How to test and debug a PHP page, How to view the source code for a web page) - How to use NetBeans to develop a PHP application (How to work with PHP projects and files, How to edit and test a PHP application , How to import and configure a PHP project) - **How to code a PHP application :** Basic PHP skills (How to embed PHP in HTML, How to code comments and statements. The six PHP data types, How to declare variables and constants) - How to get data from a request (How to use the built-in \$_GET array, How to use the built-in \$_POST array, When to use the HTTP GET and

POST methods.) - How to work with data (How to code string expressions, How to code echo statements, How to code numeric expressions, How to use the compound assignment operators, How to use some built-in functions. How to use the filter_input() function) - The Product Discount application (The user interface, The form in the HTML file, The PHP file.) - How to code control statements (How to code conditional expressions . How to code if statements, How to code while and for statements, How to pass control to another page.) - The Future Value application (The user interface, The code for the index.php file, The code for the display_results.php file) - How to use the PHP documentation (How to access the PHP manual. How to find the documentation you need) - **Practical** : PHP and MySQL Programs (PHP and MySQL Programs).

UNIT - II

LECTURE HOURS : 12

Introduction to relational databases and MySQL : PHP and MySQL Programs (PHP and MySQL Programs) - An introduction to relational databases (How a database table is organized, How the tables in a relational database are related, How the columns in a table are defined) - The SQL statements for data manipulation (How to select data from a single table, How to select data from multiple tables, How to insert, update, and delete data) - An introduction to MySQL (What MySQL provides, Two ways to work with MySQL) - How to use phpMyAdmin (How to start phpMyAdmin, How to log in, log out, and change your password, How to import and run a SQL script that creates a database, How to review the data and structure of a table. How to run SQL statements, How to create users with limited privileges) - **Practical** : PHP with a MySQL Programs (PHP with a MySQL Programs).

UNIT - III

LECTURE HOURS : 12

How to use PHP with a MySQL database : PHP for working with MySQL (How to connect to a MySQL database, How to execute SELECT statements, How to execute SELECT statements, How to execute INSERT, UPDATE, and DELETE statements ,How to use A Try/Catch statement to handle exception) - How to get data from a result set (How to work with Arrays, How to get data from the first row of a result set, How to work with all the rows of a result set,) - The Product Viewer application (The user interface, The code) - The Product Manager application (The user interface, The code) - **Practical** : PHP with a MySQL Programs (PHP with a MySQL Programs).

UNIT - IV

LECTURE HOURS : 12

How to use the MVC pattern to organize your code : How to use the MVC pattern (An introduction to the MVC pattern, How to code functions . How to redirect requests) - The Product Manager application (The user interface, The model, The controller . The view) - The Product Catalog application (The user interface, The model, The controller, The view) - **How to test and debug a PHP application :** An introduction to testing and debugging (Typical test phases for a PHP application, The three types of errors that can occur, Common PHP errors, An easy way to trace the execution of your PHP code.) - How to debug with Xdebug and NetBeans (How to set and remove breakpoints, How to step through code, How to inspect variable, How to inspect the stack trace) - **Practical :** MVC Pattern programs and Debug a PHP Application (MVC Pattern programs and Debug a PHP Application).

UNIT - V

LECTURE HOURS : 12

How to work with form data : How to get data from a form (How to get data from text boxes, password boxes, and hidden fields, How to get data from a radio button, How to get data from a check box, How to get data from an array of check boxes . How to get data from a drop-down list, How to get data from a list box, How to get data from a text area) - How to display data on a web page . (How to format special characters, How to format line breaks, How to display data with echo and print statements) - **Practical :** PHP with Forms Programs (PHP with Forms Programs).

TEXT BOOK:

T1. Joel Murach and Ray Harris, Murach's PHP and MySQL, Murach publication, 3rd edition, 2010

REFERENCES:

- R1. W. Jason Gilmore, Beginning PHP and MySQL: From Novice to Professional, Third Edition, 2008.
- R2. Luke Welling, Laura Thomson, PHP and MySQL Web Development, Pearson Publication, Fourth Edition, 2011
- R3. Michael Glass, Yann Le Scouarnec, Elizabeth Naramore, Gary Mailer, Jeremy Stolz, Jason Gerner, PHP, Apache, MySQL® Web Development, Wiley Publishing, Inc, 2004

Course Title	: PHP and MySQL (P)	Course Code	: 43E
Semester	: IV	Course Group	: DSE-II
Teaching Scheme in Hrs (L:T:P)	: 0:0:4	Credits	: 2 Credits
Map Code	: G (PRACTICAL-	Total Contact Hours	: 60 Hrs
CIA	: 40 Marks	SEE #	: 60 Marks
Programme	: B.C.A	# - Semester End Exam	

Program List :**Project 1 : Manage Products****Project 2 : Manage Technicians****Project 3 : Manage Customers****Project 4 : Register Product****Project 5 : Create Incident****Project 6 : Use a Drop - down List****TEXT BOOK:**

T1. Joel Murach and Ray Harris, Murach's PHP and MySQL, Murach publication,3rd edition,2010

REFERENCES:

- R1. W. Jason Gilmore, Beginning PHP and MySQL: From Novice to Professional, Third Edition, 2008.
- R2. Luke Welling, Laura Thomson, PHP and MySQL Web Development, Pearson Publication, Fourth Edition,2011
- R3. Michael Glass,Yann Le Scouarnec,Elizabeth Naramore,Gary Mailer,Jeremy Stolz,Jason Gerner, PHP, Apache,MySQL® Web Development,Wiley Publishing,Inc,2004

SEMESTER - V

Course Title : DATABASE MANAGEMENT SYSTEM (T)	Course Code : 53A
Semester : V	Course Group : DSC-IX
Teaching Scheme in Hrs (L:T:P) : 4:0:0	Credits : 4 Credits
Map Code : D(THEORY- APPLICATION)	Total Contact Hours : 60
CIA : 25 Marks	SEE : 75 Marks
Programme: B.Sc CS IT/BCA	# - Semester End Exam

COs.	Course Outcome (COs):	PSOs	Cl. Ses	CL
CO1	Describe the basic concepts of data and database.	PSO1 & PSO4	12	AP
CO2	Explore the use of keys and functional dependencies	PSO1 & PSO4	12	AP
CO3	Understand the structure and use of Structured Query Language and Create relation with constraints	PSO1 & PSO4	12	AP
CO4	Implement the data manipulation operations with aggregation, group by and joins	PSO1 & PSO4	12	AP
CO5	Describe the concepts of normalization and apply it into real time requirements.	PSO1 & PSO4	6	AP
CO6	Understand the concept of ER modeling, challenges of ER modeling and apply to real time requirements.	PSO1 & PSO4	6	AP

UNIT – I

LECTURE HOURS - 12

Introduction to database concepts : Introduction- (Basic building block- collection of facts,

Qualitative Vs Quantitative, Example: What do we know about the Dog?). Database - (Definition. Database Management System. Definition and purpose). Database Applications – (Enterprise , banking and Finance, characteristics of a good DB). File processing systems – (Characteristics of a good database, Data Storage relationship, Isolated, Example). Advantages over file systems – (Reduces Data redundancy and data inconsistency Data isolation, Data integrity Atomicity of operations Concurrency Security). Data Abstraction- (View of Data, University Database Example). **Relational Data Model :** Introduction – (Attributes, Domain, Example). Database Schema and Instance – (Definition Relational Schema, Relational instance Example). Physical Data Independence, Alternative Terminology – (Definition, Example). **Keys:** Introduction – (Purpose of the keys). Super Key, Candidate key – (Definition, Example). Primary key – (Points to remember for a Primary key). Secondary Key – (Definition, Example). Foreign key – (Definition, Example). Query Languages – (Procedural language Nonprocedural language Definition, Example). **Procedural language Nonprocedural language Definition, Example:** Keys- (Defining Various key from a database).

UNIT – II

LECTURE HOURS :12

Basic Structure SQL: Introduction – (Declarative language Example). Several types of SQL- (DDL, DML, Integrity, View, Transaction, Embedded SQL, Authorization). Select clause- (Syntax and Examples). From clause – (Syntax and Examples). Where clause – (Syntax and Examples). Create table – (Syntax and example). The Rename Operation- (Syntax and example). Tuple Variables – (Syntax and example). **String Operations:** Like- (Percent (%) Underscore (_)). **Display of Tuples:** order by clause- (Ascending and descending order). Set operations – (Union Operation ,Union all, Intersect Operation, Except Operation) **Assignment / Lab Exercise(s) : Data Definition:** Creating and renaming relation- (Basic of table creation and various clauses. String operations, Tuple Display- Percent (%) Underscore (_), Ascending and descending order). Set Operations – (Union Operation ,Union all, Intersect Operation, Except Operation).

UNIT – III

LECTURE HOURS : 12

Union Operation ,Unionall, Intersect Operation, Except Operation: Basic Aggregation – (Average, Sum ,Count, Maximum, Minimum). group by – (Syntax and example). Having – (Syntax and example). **Modification of the Database :** Insertion – (Syntax and example). Deletion – (Syntax and example). Updates – (Syntax and example). Syntax and example – (Standard and Other). **Join Expressions:** Join types and Join conditions – (Syntax and example). natural join operation- (Syntax and example). Outer

join- (Left right, full outer join). View Definition – (Syntax and example). **Constraints:** Not null constraint- (Syntax and Example). Unique Constraint – (Syntax and Example). The check Clause- (Syntax and Example). Primary key Constraint – (Syntax and Example). Referential Integrity - (Syntax and Example). alter table – (Add, Modify, drop table). **Assignment / Lab Exercise(s): Data Manipulation:** Aggregate and Grouping Functions – (Example) . Modification of the Database- (Example). Join Expressions – (Example). Constraints – (Example).

UNIT – IV

LECTURE HOURS : 12

Relational Database Design: Features of Good Relational Designs- (Scenarios and Examples). Introduction – FDs are domain knowledge – (DB engine will not help, no optimization, relation satisfying a dependency Vs Dependency holding on a schema). Types of FD – (Trivial, Non trivial, Completely non-trivial). Minimal FDs – (Definition, Example). Dependency Preservation – (Definition, Example). Closure and covers of set of functional dependencies- (Definition, Example). Armstrong's axioms – (Reflexive, Augmentation Transitive Decomposition, Union P pseudotransitivity). **Basic Normal forms:** First Normal Form (1NF)- (1NF: based on attributes Only). Second Normal Form (2NF) – (Prime attribute, full functional dependency and partial dependency). Third Normal Form (3NF) – (Transitive Dependency). **Higher Normal Forms :** Boyce-Codd normal (BCNF) – (Normal forms Lossless decomposition Anomalies with BCNF). Fourth normal form (4NF) – (Multi-valued dependency (MVD) MVD and lossless join). Fifth normal form (5NF) or Project-(Join normal form (PJNF)- Join dependency (JD)). Domain-Key normal form (DKNF) – (Syntax and Example). **Assignment / Lab Exercise(s): Normalization:** Normalization – (Complete Example).

UNIT – V

LECTURE HOURS : 12

The ER Model constraints: Entity Sets – (Definition and example, Relationship Sets - Definition and example). Attributes – (Definition and example, Mapping Cardinalities- Definition and example). Participation Constraints- (Definition and example). Keys – (Definition and example). What to remove – (Definition and example). **Entity- Relationship Diagrams :** Basic Structure- (Example, Mapping Cardinality- Example, ComplexAttributes- Example, Roles- Example , Non binary Relationship Sets- Example, Weak Entity Sets- Example). E-R diagram for the University Enterprise – (Example – diagram). **Reduction to Relational Schemas:** Representation of Strong Entity Sets with Simple Attributes- (Example. Representation of Strong Entity Sets with Complex Attributes – Example). Representation of Weak Entity Sets – (Example). Representation of Relationship Sets – (Redundancy of Schemas, Combination of Schemas). **Extended E-R Features:** Specialization- (Example).

Generalization – (Example). Inheritance – (Example). **Assignment / Lab Exercise(s): ER Modeling :**
ER Diagram Creation – (Complete Example).

REFERENCES:

- R1. Database System Concepts, Silberschatz-Korth-Sudarshan, 6th Edition, 2011, TMH Publications
- R2. Database System Concepts, Silberschatz-Korth-Sudarshan, 4th Edition, 2001, TMH Publications.
- R3. Fundamentals of Database Systems, RamezElmasri, Shamkant B. Navathe, 6th Edition , 2011, Pearson Education publication.
- R4. Beginning Database Solutions, Rod Stephens, 2009, Wrox Publications.

Course Title : DATABASE MANAGEMENT SYSTEM (P)	Course Code : 53P
Semester : V	Course Group: DSC-IX
Teaching Scheme in Hrs (L:T:P) : 0:0:4	Credits : 2 Credits
Map Code : M(PRACTICAL-APPLICATION)	Total Contact Hours : 60
CIA : 40 Marks	SEE : 60 Marks
Programme : B. Sc CS /B.Sc IT/BCA	# - Semester End Exam

PRACTICAL LIST

Project Allocation System.

1. Create the Project table consisting of the field's projno, location, custname and year for storing the information about the projects carried out in the organization. Define projno as the primary key column.
2. Create the Department table consisting of the field's deptno and deptname for storing the information about the departments available in the organization. Define deptno as the primary key column.
3. Create the Employee table consisting of the field's empno, empname, deptno, projno and salary for storing the information about the employees working in the organization. Define empno as the primary key column.
4. Define projno column in the Employee table as a foreign key that references the projno column in the Project table.
5. Create another table Parts consisting of the field's partno, partdesc, vndornname and cost for storing the information about the parts used in the projects. Define partno as the primary key column.
6. Create another table ProjectsParts consisting of the field's projno, partno and qty for storing the information about the quantity of the parts used in every project. Define partno and projno columns combined together as a composite primary key.
7. Define projno column in the ProjectsParts table as a foreign key that references the projno column in the Project table. Similarly define partno column in the ProjectsParts table as a foreign key that references the partno column in the Parts table.
8. Add a new field emailid to the Employee table and define a unique constraint to the newly added emailid column.
9. Modify the location column in Project table and assigning a default value 'Coimbatore' to the location

column.

10. Remove the emailid column from the Employee table using the alter statement.
11. Define a check constraint for the salary column in Employee table such that it accepts a value within the range 10000 and 70000.
12. Uses the insert statement to insert the records into all the tables created and while inserting the records validate all the constraints specified in the tables.
13. Use the update statement to increase the salary of every employee working in deptno 10 by 15%.
14. Use the delete statement to remove the information about a particular employee from Employee table.
15. Use the select statement to retrieve the required data from the various tables available in the database.

Course Title : JAVA SCRIPT AND JQUERY	Course Code : 53B
Semester : V	Course Group : DSC -X
Teaching Scheme in Hrs (L:T:P) : 5:1:0	Credits: 6 Credits
Map Code: D (THEORY – APPLICATION)	Total Contact Hours: 90
CIA: 25 Marks	SEE # : 75 Marks
Programme: BCA	#- Semester End Exam

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Understand the basic concepts of web scripting	PSO2,PSO4	8L+2T	U
CO2	Develop programs by using operators, variables	PSO2,PSO4	7L + 1T	Ap
CO3	Develop programs using control structures and arrays	PSO2,PSO4	15L+3T	Ap
CO4	Understand and Implement Functions and Events	PSO2,PSO4	15L+3T	Ap
CO5	Implement Image slide show with timer and swapping	PSO2,PSO4	15L+3T	Ap
CO6	Understand and apply JQuery concepts into a static web page	PSO2,PSO4	15L+3T	Ap

UNIT – I**LECTURE HOURS : 18**

INTRODUCTION : The components of a web application (Clients, Web server, Web browser, Networks) - **TO WEB DEVELOPMENT** : How static web pages are processed (HTTP Request and Response) - How dynamic web pages are processed (HTTP Request and Response) - How JavaScript and jQuery are used for client-side processing (Client side processing, Server side processing, Image swaps and Roll over) - **THE COMPONENTS OF A JAVASCRIPT APPLICATION** : The HTML (Explanation of content and structure of web page) - The CSS (Explanation about separating the format from content and structure of the web page) - The JavaScript (DOM Scripting, How to script the element in HTML that includes javasript) - **HOW TO INCLUDE JAVASCRIPT IN AN HTML**

DOCUMENT : Two ways to include JavaScript in the head of an HTML document (A script element in the head section that loads an external javascript file) - How to include JavaScript in the body of an HTML document (Javascript in the body of an HTML element.) - **THE JAVASCRIPT SYNTAX** : How to code JavaScript statements (The basic syntax rules) - How to create identifiers (Rules for creating identifiers.) - How to use comments (The basic syntax rules for Javascript comments.) - **HOW TO WORK WITH JAVASCRIPT DATA** : The primitive data types (Javascript's primitive datatypes) - How to declare and assign values to variables (How to declare and assign values to variables in two statements, How to declare and assign values to variables in one statements) - How to code arithmetic expressions (Javascript's arithmetic operators.) - How to use arithmetic expressions in assignment statements (Code that calculated sales tax) - How to concatenate strings and include special characters in strings (The concatenation operators for strings) - **TUTORIAL : HTML,CSS AND IDENTIFIERS** : Program (Develop the MPG application).

UNIT - II

LECTURE HOURS : 18

HOW TO CODE CONDITIONAL OPERATORS : How to use the relational operators (Relational operators, conditional expression, isNaN method) - How to use the logical operators (The logical operators) - **HOW TO CODE BASIC CONTROL STATEMENTS** : How to code if statements (explanation) - How to code while and do-while loops (The Syntax of a while loop, The Syntax of a do-while loop,) - How to code for loops (The Syntax of a for statment) - **THE THREE ILLUSTRATIVE APPLICATIONS** : The enhanced Miles Per Gallon application (The HTML and Javascript for the application) - The Future value application (The HTML and Javascript for the application) - The Enhanced test scores application (The Javascript in the dead section of the HTML File) - **HOW TO WORK WITH ARRAYS** : How to create and use the arrays (The syntax for creating array) - How to use for loops to work with arrays (Code that puts the numbers 1 through 10 into an array) - **THE TEST SCORES APPLICATION WITH AN ARRAY** : The user interface, The Javascript (The javascript for the enhanced Test Scores application) - **TUTORIAL : OPERATORS, CONDITIONAL STATEMENTS** : Program (JavaScript can validate input by giving numbers from 1 to 9. program to accept input as 0-9 numbers. if the input is two digit ,display an error in an alert window)

UNIT - III

LECTURE HOURS : 18

HOW TO USE OBJECTS TO WORK WITH DATA : How to use the window and document objects (Another methods of the window object that displays a dialog box) - How to use Textbox and Number

objects (Explanation-Method, Properties of textbox and number object) - How to use Date and String objects (Explanation-Method, Properties of date and string object) - **HOW TO USE FUNCTIONS** : How to create and call a function expression (The syntax for a function expression) - How to create and call a function declaration (A function that uses local variable named tax, A function that uses global variable named tax) - When and how to use local and global variables (A function that uses a local variable named tax) - When and how to use strict mode (The strict mode directive) - **HOW TO HANDLE EVENTS** : How to attach an event handler to an event (Explanation of common events) - How to use an onload event handler to attach the other event handlers (The web browser after the Email Address has been changed) - **TWO ILLUSTRATIVE APPLICATIONS** : The Miles per Gallon Application (The Miles Gallon application in a browser) - The Email List Application (The Email list application in a browser, The JavaScript for the Email list application) - **HOW TO TEST AND DEBUG A JAVASCRIPT APPLICATION** : Three types of error that can occur (The Future value application with a logic error) - Common Javascript errors (Common syntax errors) - How to plan the test runs (The Future Value application as its tested with valid data. The Future Value application as its tested with invalid data.) - How to use top down coding and testing to simplify debugging (The user interface for a Future Value application) - **TUTORIAL : FUNCTIONS, EVENTS, ARRAYS** : Program (Develop the Sales Tax Calculator) -

UNIT - IV

LECTURE HOURS : 18

HOW TO WORK WITH LINKS AND IMAGES : How to cancel the default action of an event (Common HTML elements that have default actions for the click event) - How to preload the images (How to preload an image with the Image object) - **THE IMAGE SWAP APPLICATION** : The HTML and CSS (The user interface for the Image Swap application) - The Javascript (The Swap.js file) - **HOW TO USE TIMERS** : How to use a one-time timer (Two methods for working with a timer that calls a function once) - How to use an interval timer (Two methods for working with a timer that calls a function repeatedly) - **THE SLIDE SHOW APPLICATION** : The HTML and CSS (The user interface for the Slide Show application) - The Javascript (The Javascript for the Slide Show application) - **TUTORIAL : LINKS, IMAGES, SLIDE SHOW, IMAGESWAP** : Program (Sketch the user interface for the Image Swap application) -

UNIT - V

LECTURE HOURS :18

INTRODUCTION TO JQUERY : What is jQuery is (What jQuery offers) - How jQuery can simplify Javascript development (The FAQ application in a browser) - **THE BASIS OF JQUERY PROGRAMMING :** How to include of jQuery in your web pages (The jQuery download page at jquery.com/download) - How to code jQuery selectors (The syntax for a jQuery selector) - How to call jQuery methods (The syntax for calling a jQuery method) - How to use jQuery event methods (The syntax for a jQuery event method) - **THE EMAIL LIST APPLICATION IN JQUERY :** The user interface and HTML (The user interface for the Email List application) - The jQuery (The jQuery for the Email List application (email_list.js)) - **A WORKING SUBSET OF SELECTORS , METHODS AND EVENT METHODS :** The most useful selectors (A summary of the most useful jQuery selectors) - The most useful methods (A summary of the most useful jQuery methods) - The most useful event methods (A summary of the most useful jQuery event methods) - Other event methods that you should be aware of (Other event methods that you should be aware of) - **THREE ILLUSTRATIVE APPLICATIONS :** The FAQ's application in jQuery (The FAQ's application in browser) - The Image application in jQuery (The user interface for the Image Swap application) - The Image Rollover application in jQuery (Two Images with second Image rolled over) - **HOW TO HANDLE EXCEPTIONS :** How to use try-catch statements (The syntax for a try-catch statement, Two properties of Error objects) - How to create and throw Error objects (The syntax for creating a new Error object, The syntax for throw statement) - **TUTORIAL : JQUERY SELECTORS, METHODS :** Program (Add all the effects to the Image Gallery app) –

TEXT BOOK:

T1. “JavaScript and jQuery” Mary Delamater and Zak Ruvalcaba, Mike Murach's Publications, 3rd edition, July2017.

REFERENCE BOOKS:

R1.” Web Development with jQuery®” Richard York, WROX publications, 2015.

R2. “High Performance JavaScript”, Nicholas C. Zakas,O'reilly, publications, 2010

R3. “JavaScript and JQuery: Interactive Front-End Web Development” Jon Duckett, WILEY & SONS Publications, 2014.

Course Title : ANDROID PROGRAMMING I	Course Code : 53C
Semester : V	Course Group : DSE-
Teaching Scheme in Hrs (L:T:P) : 4:0:0	Credits : 4
Map Code : D (THEORY-PROGRAMMING)	Total Contact Hours : 60
CIA : 25 Marks	SEE # : 75
Programme : B.Sc. CS/IT/BCA	5 - Semester End Exam

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Explain and illustrate the concepts of Variables , Constants, Control Statements, Exceptions and Classes	PSO4	12	Ap
CO2	Define and apply Inheritance, Interfaces, and Arrays	PSO4	12	Ap
CO3	Define and illustrate user interface widgets and layouts	PSO1	12	Ap
CO4	Describe the importance of android and demonstrate android app eclipse	PSO4	12	Ap
CO5	Perform Interface functions in Android	PSO4	12	Ap
CO6	Apply Java code with events for Android App	PSO4	12	Ap
CO7	Demonstrate to test and debug an android app	PSO4	12	Ap

UNIT - I**LECTURE HOURS : 12**

Basic skills for working with data: The eight primitive data types-(Description, Technical Notes). How to declare and initialize variables- (How to declare and initialize variables in two Statements, How to declare and initialize variables in One Statement, Description, Naming Conventions). How to declare and initialize constants-(Description, Naming Conventions). How to code assignment statements and arithmetic expressions- (Arithmetic Operators, Examples of Simple Assignment Statements,

Description). How to use the shortcut assignment operators-(Assignment Operators, Statement that use the same variable on both sides of the equals sign, Statement that use the shortcut operators to get the same results, Description). How to work with the order of precedence- (The order of precedence for arithmetic operations, Example 1:Code that calculates a discounted price, Example 2:Code that calculates the current value of monthly investment, Example 3:Prefixed and postfix increment and decrement operator, Description). How to work with casting-(How implicit casting works, How to code an explicit cast, How to cast between char and int types,Description).**How to code Boolean expressions:** How to compare primitive data types-(Relational operators, Examples of Boolean expressions, Description). How to compare strings-(Two methods of the String class, Expression that compare two string values, Code that tests whether two string refer to the same object, Description). How to use the logical operators-(Logical Operators, Examples, Description). **How to code if/else and switch statements:** How to code if/else statements- (The syntax of the if/else statement, Example 1: An if statement with else if and else clauses, Example 2:An if statement that contains two blocks of code, Example 3: Nested if statements , Description). How to code switch statements-(The syntax of the switch statement , Example 1:A switch statement that uses an integer, Example 2: A switch statement that uses a string, Description Example 3: A Switch that fails through case labels, Description). An enhanced version of the Invoice application-(The console, The Code). **How to code loops:** How to code while and do-while loops-(The Syntax of the while loop, The syntax of the do-while loop, Description). How to code for loops-(The syntax of the for loop, Example 1:A for loop that stores the numbers 0 through 4 in a string with a single statement, Example 2:A for loop that adds the numbers 8,6,4 and 2, Example 3:A for loop that calculates a future value, Description). The Future Value application- (The console, The Code). How to code nested loops-(The console,, Nested loops that print a table of future values). **How to code break and continue statements:** How to code break statements-(The syntax of the break statement , Example 1:A break statement that exists the inner loop, The syntax of the labeled break statement, The structure of the labeled break statement, Example 2:A labeled break statement that exists the outer loop, Description). How to code continue statements-(The Syntax of the continue statement, Example 1:A continue statement that jumps to the beginning of a loop, The syntax of the labeled continue statement, The structure of the labeled continue statement, Example 2: A labeled continue statement that jumps to the beginning of the outer loop, Description). **How to code and call static methods:** How to code static methods-(Description). How to call static methods-(The basic syntax for the coding a static method, The syntax for calling a static method that's in the same class, Description). The Future Value application with a static methods-(The Code). How to handle

exceptions- How exceptions work(Some of the classes in the Exception hierarchy, The console after an Input MismatchException has been thrown, Four methods that might throw an exception, Description). How to catch exceptions-(The syntax for the try statement, The two ways to import the InputMismatchException, Console output, Description). The Future Value application with exception handling- (The code for the Future Value application with exception handling). **How to validate data:** How to prevent exceptions from being thrown -(Methods of the scanner class you can use to validate data, Example 1:Code that prevents an InputMismatchException, Example 2:Code that prevents a NullPointerException, Description). How to validate a single entry-(Code that gets a valid double value within a specified rang, Description). How to use generic methods to validate an entry-(A method that gets a valid numeric format, A method the checks for a valid numeric range, Code that uses these methods to return two valid double values, Description). **An introduction to classes:** How classes can be used to structure an application-(The architecture of a three- tiered application, Description). How encapsulation works-(A class diagram for the product class, Description, UML diagraming notes). The relationship between a class and its objects-(The relationship between a class and its objects, Description). **How to code a class that defines an object :**How to use NetBeans to create a new class-(The Dialog box for creating a new java class, The code that's generated for the Product class, Description). The code for the Product class-(The Product Class). How to code instance variables-(The syntax for declaring instance variables, Examples, Where you can declare instance variables, Description). How to code constructors-(The syntax for coding constructors, Example 1:A Constructor that assigns default values, Example 2:A custom Constructor with three parameters , Example 3:Another way to code that constructor Example 4: A constructor with one parameter, Description). How to code methods-(The Syntax for coding a method, Example 1: A method that doesn't accept parameters or return data. Example 2:A get method that returns a String, Example 3: A get method that returns a double value, Example 4:A custom get method, Example 5:A set method, Example 6:Another way to code a set method, description). How to overload methods-(Example 1: A method that accepts one argument, Example 2:An overloaded method that provides a default value, Example 3: An overloaded method with two arguments, Code that calls these methods, The Console, Description). How to use the this keyword-(The syntax for using the This keyword, Example 1: How to refer to instance variables, Example 2: How to refer to methods, Example 3:How to call a constructor, Example 4:How to send the current object t a method, Example 5: How to send the current object to a static method, Description). How to use NetBeans to work with classes-(The NetBeans window for the Product Application, The dialog box for generating get and set methods, Description). **How to create and use**

an object: How to create an object-(How to create an object in two statements, How to create an object in one statement, Description). How to call the methods of an object-(How to call a method, Description). How primitive types and reference types are passed to a method-(Example 1:A primitive type that's passed to a method, Example 2:An object that type's passed to a method, Description). A ProductDB class that creates a Product object-(The ProductDB class). A ProductApp class that uses a Product object-(The Console, The ProductApp Class). **How to code and use static fields and methods:** How to code static fields and methods-(How to declare static fields, Example 1:A Class that contains a static constant and a static method, Example 2: The Product class with a Static variable and a static method, Description). How to call static fields and methods-(The syntax for calling a static field or method, How to call static fields, How to call static methods, A statement that calls a static field and a static method, Description). How to code a static initialization block-(Description). When to use static fields and methods-(The syntax for coding a static initialization block, A class that uses a static initialization block, Description). **Practical:** Test the Invoice application-(Aim , Algorithm and Code). Modify the Test Score Application-(Aim , Algorithm and Code). Create a new Application-(Aim , Algorithm and Code). Test the Future value Application-(Aim , Algorithm and Code). Enhance the Invoice Application-(Aim , Algorithm and Code). Enhance the Test Score Application-(Aim , Algorithm and Code). Add validation to the Invoice Application-(Aim , Algorithm and Code). Add validation to the Test Score Application-(Aim , Algorithm and Code). Enhance the Future Value Application-(Aim , Algorithm and Code).

UNIT - II

LECTURE HOURS :12

An introduction to inheritance: How inheritance works-(How inheritance works, Description). How the Java API uses inheritance-(The inheritance hierarchy for swing forms and controls, Description). How the Object class works-(The Object class, Methods of the Object Class, Description). How to use inheritance in your applications-(Business classes for a Product Maintenance application, Description). **Basic skills for working with inheritance:** How to create a superclass-(Access Modifiers, An annotation for overriding a method, The code for the Product superclass, Description). How to create a subclass-(The syntax for creating subclasses, the code for a Book subclass, Description). How polymorphism works-(Three versions of the toString method, Code that uses the overridden methods, Description). **An introduction to interfaces:** A simple interface-(Example 1: A printable interface that defines a print method, Example 2: A Product class that implements the Printable interface, Example 3:Code that uses the print method of the Product class, Description). Interfaces compared to abstract

classes-(An abstract class compared to an interface, Example 1:APrintable interface, Example 2: A Printable abstract class, Advantages of an abstract class, Advantages of an interface). Some interfaces of the Java API-(Some interfaces in the java.lang package, Some interfaces in the java.util and java.awt.event packages, Description). **How to work with interfaces:** How to code an interface-(The syntax for declaring an interface, Example 1:An interface that defines one method, Example 2:An interface that defines three methods, Example 3:An interface that defines constants, Example 4:A tagging interface with no members, Description). How to implement an interface-(The syntax for implementing an interface, A class that implements two interfaces, Description). How to inherit a class and implement an interface-(The syntax for inheriting a class and implementing an interface, A book class that inherits Product and implements Printable, Description). How to inherit a class and implement an interface-(The syntax for inheriting a class and implementing an interface, A book class that inherits Product and implements Printable, Description.). How to use an interface as a parameter-(Example 1:A method that accepts a Printable object, Example 2:Code that passes a product object to the method, Example 3:Code that passes a Printable object to the method, Description). How to use inheritance with interfaces-(The syntax for declaring an interface that inherits other interfaces, Example 1:A ProductReader interface, Example 2: A ProductWriter interface, Example 3:A ProductConstants interface, Exampe 4: A ProductDAO interface that inherits these three interfaces, Description). How to use NetBeans to work with interfaces-(A class that implements the ProductDAO interface, The code that's generated by NetBeans, Description). **Basic skills for working with arrays:** How to create an array-(The syntax for declaring and instantiating an array, Examples of array declarations, other Examples, Description). How to assign values to the elements of an array-(The syntax for referring to an element of an array, Examples that assign values by accessing each element, The syntax for creating an array and assigning values in one statement, Examples that create an array and assign values in one statement, Description.). How to use for loops with arrays-(The syntax for getting the length of an array, Example 1: Code that puts the numbers 0 through 9 in an array, Example 2: Code that prints an array of prices to the console, Example 3:Code that computes the average of the array of prices, Example 4:Another way to compute the average in a for loop, Description). How to use enhanced for loops with arrays-(The syntax of the enhanced for loop, Example 1:Code that prints an array of prices to the console, Example 2:Code that computes the average of the array of prices, Description). **Practical:** Look at a class that inherits the JFrame class-(Aim , Algorithm and Code). Create and work with interfaces-(Aim , Algorithm and Code). Use one dimension array-(Aim , Algorithm and Code). Use a

rectangular array-(Aim , Algorithm and Code). Sort an array of user– defined objects-(Aim , Algorithm and Code). Work with a deck of cards-(Aim , Algorithm and Code).

UNIT - III

LECTURE HOURS : 12

Android overview: Types of devices-(Readers , cameras, Home automation systems ,Home appliances, vehicle systems, game consoles). Types of Apps-(Category , functionality). A brief history-(Open Handset Alliance , Android open Source Project(AOSP)). Versions-(Android versions and descriptions). System Architecture-(Android stack ,Dalvik virtual machine(DVM) , Java virtual machine(JVM)). How Apps are compiled and run-(Android system architecture , Integrated Development Environment(IDE) , Android Virtual Device(AVD),Android Debug Bridge(ADB)).

A simple Android app: The user interface-(The tip calculator app with the Soft keyboard displayed). The XML for the user interface-(The activity_tip_calculator.xml file). The XML for the display text-(The activity_tip_calculator.xml file). The java source code-(The activity_tip_calculator.java file). The Android Manifest-(The Androidmanifest.xml file).

How to develop the user interface: The tip calculator APP-(The use interface for the Tip Calculator). How to work with a layout-(The default layout for an activity in a new project). How to add widgets to a layout -(A layout after some widgets have been added to it). How to set the display text-(A layout after the text has been set for the widgets). How to work with the strings.xml file-(The strings.xml file for the Tip Calculator app). How to set properties-(A layout after some properties of the widgets have been set). Common properties-(Common properties for layouts , Common properties for widgets, predefined values for setting height and width , common units of measurement). The XML for the user Interface-(The XML for the user interface).

How to write the Java code: How to work with an Activity-(The default java code for an activity). How to get references to widgets-(An activity that gets references to the widgets). How to get and set text for widgets-(getText method , setText method). How to handle the click Event-(Import the interface for the listener,Implement the interface for the listener, Implement the interface for the listener). The life cycle of an activity-(Resumed , paused , stopped , created state , started state). How to save and restore values-(How to import the SharedPreferences class and editor class , How to set up the instance variable, how to use onPause method to save values , how to use onResume method to restore values). How to use the documentation for the android API-(use the documentation for the Activity class). The java code for the APP-(Java code).

Practical: Create the invoice Total App-(XML Code and Java Code). Use the documentation for the Android API-(XML Code and Java Code).

UNIT - IV**LECTURE HOURS : 12**

Basic skills for testing and debugging: Typical test phases-(The Tip calculator with logical error). How to check the layout-(Graphical Layout Editor for the Android 2.2 platform). The three types of errors-(Syntax errors , runtime errors , exceptions , logic errors). How to handle run time errors-(The error that's displayed when an app crashes). **How to trace Code execution:** How to use LogCat logging-(Eclipse with the LogCat view displayed, A few methods of the Log class, How to use the Log Class, Description). How to use toasts-(A toast displayed in an emulator, Two methods of the toast class, Two constant of the toast class, How to display a toast). **How to use debugger:** How to set and remove breakpoints-(The Java perspective with a breakpoint, The Confirm perspective Switch dialog box, Description). How to step through code-(Step Into button, Resume button). How to inspect variables-(Variable named this). How to inspect stack trace-(The Debug perspective, Description). How to configure step filters-(The step filtering preferences, Common packages to add to step filtering, Description). **How to configure your emulators:** How to add an emulator for an old phone-(The device definition for a phone with a hard keyboard and DPad, How to create a new device definition, Description, An emulator that supports a hard keyboard and DPad, How to create an emulator, Description). How to work with an emulator for an old phone-(The soft keyboard on an emulator for an old phone, description). How to add an emulator for a tablet-(An emulator for a tablet, description). **An introduction to layouts and widgets:** A summary of layouts -(RelativeLayout , LinearLayout , Table Layout , FrameLayout , AbsoluteLayout , GridLayout). A summary of widgets-(TextView , EditText , Button ,CheckBox). The view hierarchy-(ViewGroup , TextView , ProgressBar , ImageView). **How to work with layouts:** How to use a linear layout-(A linear layout with vertical orientation and two buttons). How to use a table layout-(A table layout with two rows and four columns). How to use frame layout -(A frame layout that displays an image behind some text). How to nest layout-(Nested linear layouts). How to provide a landscape layout-(The location of the xml files ,the xml for landscape orientation). **How to work with widgets:** How to use editable text views-(The soft keyboard for an editable text view for an email address). How to use check boxes-(A common xml attribute for check boxes , two common java methods for check boxes). How to use radio buttons-(Three radio buttons in a radio group with vertical orientation , Three radio buttons in a radio group with horizontal orientation). How to use spinners-(The xml code,the array in the strings.xml file). How to use seek bars-(Two common XML attributes for seek bars , two common java methods for seek bars). How to display images-(Two attributes of an ImageView widget, Four qualifiers for the drawable folder). How to show and hide widgets-(Description). How to add scroll bar-(Description). **Practical:** Test and Debug the

Tip Calculator app-(XML Code and Java Code). Modify the Layout for the Tip Calculator App-(XML Code and Java Code).

UNIT - V

LECTURE HOURS : 12

A Summary of listeners: High –Level Events-(Use the current class as the listener). Low Level Events-(Import the interface for the Listener , Use the current class as the listener , Use a separate named class as the listener). **Four techniques for handing events:** How to use the current class as the listener-(use an anonymous class as the listener). How to use a named class as the listener-(use an inner anonymous class as the listener). How to use an anonymous class as the listener-(Description). How to use an inner anonymous class as the listener-(Description). When to use each technique-(Description). **How to handle high-level events:** How to handle events for check boxes and radio buttons-(A check box, An event handler for check box, Another event handler for check box, A method of the View class, Description). How to handle events for radio groups-(Three radio buttons in a group, An event handler for a radio button, Another event handler for radio button, description). How to handle events for spinners-(A spinner, An event handler for a spinner, description). How to handle events for seek bars-(A seek bar and a label , An event handler for a seek bar, description). **How to handle low-level events:** How to handle key events-(An event handler for the key event, some constants from the KeyEvent class, description). How to handle Touch events-(An event handler for a Touch event, some constants of the MotionEvent class, some methods of the MotionEvent class, description). **The Tip Calculator App:** The User Interface-(Description). The Java Code for the activity-(Description). **Practical:** Use anonymous classes for the event listeners-(XML Code and Java Code). Improve the Listeners for the key Events-(XML Code and Java Code).

TEXT BOOK:

T1. Training and Reference Murach's Java programming by Joel Murach , Mike Murach&Associates ,Inc. 2013

REFERENCE BOOKS:

R1. Beginning Android Application Development by Wei-Meng Lee and Wiley Publishing, Inc. 2011

R2. Android Programming Tutorials by Mark L. Murphy and CommonsWare, LLC., 2011

R3. Android Apps for Absolute Beginners Edition:2 APress Wallace Jackson(2013)

Course Title : ANDROID PROGRAMMING I (P)	Course Code: 53Q
Semester : V	Course Group: DSE-III
Teaching Scheme in Hrs (L:T:P) : 0:0:4	Credits: 2 Credits
Map Code: M (PRACTICAL-PROGRAMMING)	Total Contact Hours : 60
CIA : 40 Marks	SEE #: 60 Marks
Programme : B.Sc. CS/IT/BCA	5 - Semester End

LIST OF PROGRAMMS

- 1) Create a java program to Test the Invoice application
- 2) Create a java program to Test the Future value Application
- 3) Using array concept develop java program to work with a deck of cards
- 4) Develop a java program to modify and test the invoice application
- 5) Create a java program to enhance the Invoice Application
- 6) How to add validation to the Invoice Application
- 7) Write a program to work with Classes and interface
- 8) Design android layout for invoice application, add discount and coupon discount for the invoice total
- 9) Design android program to Use anonymous classes for the event listeners
- 10) Create android layout for invoice application to generate GST bill.

TEXT BOOK:

T1. Training and Reference Murach's Java programming by Joel Murach , Mike Murach&Associates ,Inc. 2013

REFERENCE BOOKS:

- R1. Beginning Android Application Development by Wei-Meng Lee and Wiley Publishing, Inc. 2011
- R2. Android Programming Tutorials by Mark L. Murphy and CommonsWare, LLC., 2011
- R3. Android Apps for Absolute Beginners Edition:2 APress Wallace Jackson(2013)

Course Title : EDC -JAVASCRIPT AND JQUERY (T)	Course Code : 5ED
Semester : V	Course Group : EDC
Teaching Scheme in Hrs (L:T:P) : 5:1:0	Credits : 6 Credits
Map Code: D (THEORY – APPLICATION)	Total Contact Hours: 90
CIA : 25 Marks	SEE # : 75 Marks
Programme : BCA	# - Semester End Exam

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Understand the basic concepts of web scripting	PSO2,PSO4	10	U
CO2	Develop programs by using operators, variables	PSO2,PSO4	8	AP
CO3	Develop programs using control structures and arrays	PSO2,PSO4	18	AP
CO4	Understand and Implement Functions and Events	PSO2,PSO4	18	AP
CO5	Implement Image slide show with timer and swapping	PSO2,PSO4	18	AP
CO6	Understand and apply JQuery concepts into a static web page	PSO2,PSO4	18	AP

UNIT - I**LECTURE HOURS : 18**

INTRODUCTION : The components of a web application (Clients, Web server, Web browser, Networks) - **TO WEB DEVELOPMENT** : How static web pages are processed (HTTP Request and Response) - How dynamic web pages are processed (HTTP Request and Response) - How JavaScript and jQuery are used for client-side processing (Client side processing, Server side processing, Image swaps and Roll over) - **THE COMPONENTS OF A JAVASCRIPT APPLICATION** : The HTML (Explanation of content and structure of web page) - The CSS (Explanation about separating the format from content and structure of the web page) - The JavaScript (DOM Scripting, How to script the element in HTML that includes javascript) - **HOW TO INCLUDE JAVASCRIPT IN AN HTML DOCUMENT** : Two ways to include JavaScript in the head of an HTML document (A script element in the head section that loads an

external javascript file) - How to include JavaScript in the body of an HTML document (Javascript in the body of an HTML element.) - **THE JAVASCRIPT SYNTAX** : How to code JavaScript statements (The basic syntax rules) - How to create identifiers (Rules for creating identifiers.) - How to use comments (The basic syntax rules for Javascript comments.) - **HOW TO WORK WITH JAVASCRIPT DATA** : The primitive data types (Javascript's primitive datatypes) - How to declare and assign values to variables (How to declare and assign values to variables in two statements, How to declare and assign values to variables in one statements) - How to code arithmetic expressions (Javascript's arithmetic operators.) - How to use arithmetic expressions in assignment statements (Code that calculated sales tax) - How to concatenate strings and include special characters in strings (The concatenation operators for strings) - **TUTORIAL : HTML,CSS AND IDENTIFIERS** : Program (Develop the MPG application) –

UNIT - II

LECTURE HOURS : 18

HOW TO CODE CONDITIONAL OPERATORS : How to use the relational operators (Relational operators, conditional expression, isNaN method) - How to use the logical operators (The logical operators) - **HOW TO CODE BASIC CONTROL STATEMENTS** : How to code if statements (explanation) - How to code while and do-while loops (The Syntax of a while loop, The Syntax of a do-while loop,) - How to code for loops (The Syntax of a for statment) - **THE THREE ILLUSTRATIVE APPLICATIONS** : The enhanced Miles Per Gallon application (The HTML and Javascript for the application) - The Future value application (The HTML and Javascript for the application) - The Enhanced test scores application (The Javascript in the dead section of the HTML File) - **HOW TO WORK WITH ARRAYS** : How to create and use the arrays (The syntax for creating array) - How to use for loops to work with arrays (Code that puts the numbers 1 through 10 into an array) - **THE TEST SCORES APPLICATION WITH AN ARRAY** : The user interface, The Javascript (The javascript for the enhanced Test Scores application) - **TUTORIAL : OPERATORS, CONDITIONAL STATEMENTS** : Program (JavaScript can validate input by giving numbers from 1 to 9. program to accept input as 0-9 numbers. if the input is two digit ,display an errorin an alert window) –

UNIT - III

LECTURE HOURS : 18

HOW TO USE OBJECTS TO WORK WITH DATA : How to use the window and document objects (Another methods of the window object that displays a dialog box) - How to use Textbox and Number objects (Explanation-Method, Properties of textbox and number object) - How to use Date and String objects (Explanation-Method, Properties of date and string object) - **HOW TO USE FUNCTIONS :** How to create and call a function expression (The syntax for a function expression) - How to create and call a function declaration (A function that uses local variable named tax, A function that uses global variable named tax) - When and how to use local and global variables (A function that uses a local variable named tax) - When and how to use strict mode (The strict mode directive) - **HOW TO HANDLE EVENTS :** How to attach an event handler to an event (Expalnation of common events) - How to use an onload event handler to attach the other event handlers (The web browser after the Email Address has been changed) - **TWO ILLUSTRATIVE APPLICATIONS :** The Miles per Gallon Application (The Miles Gallon application in a browser) - The Email List Application (The Email list application in a browser, The JavaScript for the Email list application) - **HOW TO TEST AND DEBUG A JAVASCRIPT APPLICATION :** Three types of error that can occur (The Future value application with a logic error) - Common Javascript errors (Common syntax errors) - How to plan the test runs (The Future Value application as its tested with valid data. The Future Value application as its tested with invalid data.) - How to use top down coding and testing to simplify debugging (The user interface for a Future Value application) - **TUTORIAL : FUNCTIONS, EVENTS, ARRAYS :** Program (Develop the Sales Tax Calculator).

UNIT - IV

LECTURE HOURS : 18

HOW TO WORK WITH LINKS AND IMAGES : How to cancel the default action of an event (Common HTML elements that have default actions for the click event) - How to preload the images (How to preload an image with the Image object) - **THE IMAGE SWAP APPLICATION :** The HTML and CSS (The user interface for the Image Swap application) - The Javascript (The Swap.js file) - **HOW TO USE TIMERS :** How to use a one-time timer (Two methods for working with a timer that calls a function once) - How to use an interval timer (Two methods for working with a timer that calls a function repeatedly) - **THE SLIDE SHOW APPLICATION :** The HTML and CSS (The user interface for the Slide Show application) - The Javascript (The Javascript for the Slide Show application) - **TUTORIAL : LINKS,**

IMAGES, SLIDE SHOW, IMAGESWAP : Program (Sketch the user interface for the Image Swap application) -

UNIT – V

LECTURE HOURS : 18

INTRODUCTION TO JQUERY : What is jQuery is (What jQuery offers) - How jQuery can simplify Javascript development (The FAQ application in a browser) - **THE BASIS OF JQUERY PROGRAMMING** : How to include of jQuery in your web pages (The jQuery download page at jquery.com/download) - How to code jQuery selectors (The syntax for a jQuery selector) - How to call jQuery methods (The syntax for calling a jQuery method) - How to use jQuery event methods (The syntax for a jQuery event method) - **THE EMAIL LIST APPLICATION IN JQUERY** : The user interface and HTML (The user interface for the Email List application) - The jQuery (The jQuery for the Email List application (email_list.js)) - **A WORKING SUBSET OF SELECTORS , METHODS AND EVENT METHODS** : The most useful selectors (A summary of the most useful jQuery selectors) - The most useful methods (A summary of the most useful jQuery methods) - The most useful event methods (A summary of the most useful jQuery event methods) - Other event methods that you should be aware of (Other event methods that you should be aware of) - **THREE ILLUSTRATIVE APPLICATIONS** : The FAQ's application in jQuery (The FAQ's application in browser) - The Image application in jQuery (The user interface for the Image Swap application) - The Image Rollover application in jQuery (Two Images with second Image rolled over) - **HOW TO HANDLE EXCEPTIONS** : How to use try-catch statements (The syntax for a try-catch statement, Two properties of Error objects) - How to create and throw Error objects (The syntax for creating a new Error object, The syntax for throw statement) - **TUTORIAL : JQUERY SELECTORS, METHODS** : Program (Add all the effects to the Image Gallery app).

TEXT BOOK:

T1. "JavaScript and jQuery" Mary Delamater and Zak Ruvalcaba, Mike Murach's Publications, 3rd edition, July2017.

REFERENCE BOOKS:

R1." Web Development with jQuery®" Richard York, WROX publications, 2015.

R2. "High Performance JavaScript", Nicholas C. Zakas,O'reilly, publications, 2010

R3. "JavaScript and JQuery: Interactive Front-End Web Development" Jon Duckett, WILEY & SONS Publications, 2014.

SEMESTER VI

Course Title : NOSQL DATABASE (T)	Course Code : 63A
Semester : VI	Course Group : DSC-XI
Teaching Scheme in Hrs (L:T:P) : 4:0:0	Credits : 4 Credits
Map Code: D (THEORY-APPLICATION)	Total Contact Hours : 60
CIA : 25 Marks	SEE # : 75 Marks
Programme : BSc.CS/BCA/BSc IT	# - Semester End Exam

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Discuss the basic concepts of MongoDB and Databases	PSO1	12	U
CO2	Explain the concepts of MongoDB architecture	PSO1	10	U
CO3	Illustrate MongoDB CRUD operation using insert, find, update and delete operation	PSO1	7	AP
CO4	Execute Text search using MongoDB	PSO1	5	AP
CO5	Implement Replication using Replica set	PSO1, PSO2	12	AP
CO6	Execute Sharding using shard and non-shard collection	PSO1, PSO2	14	AP

UNIT - I

LECTURE HOURS:12

Introduction to No SQL Database: Unstructured data-(What is No SQL Database, Difference between SQL and NO SQL Database, Database), **Introduction to MongoDB:** What is MongoDB-(Database, collection, Document), Document oriented database-(Document), Features of MongoDB-(Ad hoc queries, indexing, replication, duplication of data, load balancing, map reduce, schema less, java script, high performance), NoSql Databases-(NoSql Database, Advantages), Common terms in MongoDB-(_id, collection, cursor, Database, Document, Field, JSON), Difference between MongoDB & RDBMS-(Table, row, column, joins Sql, Nosql),

MongoDB Datatypes-(String, integer, Boolean, Double, Min/Max keys, Arrays, object, Null, Symbol, Date), **Databases and collections:**Views-(Create view, view creation, shared view, drop a view, modify a view, supported operation), Capped collections-(Overview, behavior, restrictions and recommendations, procedures), **Documents:** Structure-(Field name, field value limit), Dot notation-(Arrays, embedded documents), Document limitation-(Document size limit, field order, _id field), Other uses of the document structure-(Query filter documents, update specification documents, index specification documents).

UNIT - II

LECTURE HOURS:12

MongoDB Architecture:The Nexus Architecture-(Expressive query language & secondary Indexes, Strong consistency, Enterprise Management and Integrations, Flexible Data Model, Scalability and Performance, Always-On Global Deployments), **Install MongoDB:** Installing MongoDB on windows-(MongoDB package, Create a data directory, Execute MongoDB), **The mongo shell:** Configure the mongo shell-(Customize the prompt, use an external editor in the mongo shell, change the mongo shell batch size), Access the mongo shell help-(Command line help, shell help, database help, collection help, cursor help, wrapper object help), Write scripts for the mongo shell-(Opening the new connections, difference between interactive and scripted mango, scripting), Data types in the mongo shell-(Types, check types in the mango shell). Display number of operations in the Mongo Shell, Customize prompt to Display database and hostname in the Mongo Shell).

UNIT - III

LECTURE HOURS:12

MongoDB CRUD operation: Insert Document-(Insert methods), Query Document-(Nested documents, query on array, embedded documents, project fields to return from query, null or missing fields, iterate a cursor in a mongo shell), Update Documents-(Update methods), Delete Documents-(Delete methods), **Bulk write operations:** Overview-(Db.collection.bulk write()),Ordered vs unordered operations-(Ordered bulk write, unordered bulk write()),Bulkwrite method()-(Insertone(), updateone(), updatemany(), replaceone(), deleteone(), deletemany()),**SQL to MongoDB Mapping chart:** Terminology and concepts-(Terms and concepts of SQL and MongoDB), Executables-(Executables), Examples-(Examples), Insert a Single Document-(Insert a single document into an inventory collection. The inventory collection has the following fields: item, quantity, tags and size, Insert a multiple documents into an inventory collection. The inventory collection has the following fields: item, quantity, tags and size).

UNIT - IV**LECTURE HOURS:12**

Text search: Text indexes-(Text indexes), Text search operators-(Query frame work, aggregation frame work), Text search in the aggregation pipeline-(Restrictions, text score, calculate the total views for articles that contains a word, return results sorted by text search score, match on text score, specify a language for text search), **Indexes:** Single field indexes-(Create an ascending index on a single field, create an index on an embedded field, create an index on embedded document), Compound indexes-(Multi key indexes, text indexes), Hashed indexes-(Index properties, index bulk operations), Index intersection-(Index prefix intersection, index intersection and compound indexes, index intersection and sort), Manage indexes-(View existing indexes, remove indexes, modify an index, rebuild indexes).

UNIT V**LECTURE HOURS:12**

Replication: Replica set primary-(Replica set secondary members), Replica set arbiter-(Example, security),**Sharding:** Shared cluster-(Shared cluster),Shared keys -(Shared keys),Chunks-(Chunks), Advantages of sharding-(Read/write, storage capacity, high availability), Shared and non-shared collections-(Shared, non-shared collections), Connecting to a shared cluster-(Mongos, shared cluster),

REFERENCES:

1. Kristina chodorow “MongoDB, The definitive guide”, 2nd Edition, O’Reilly Publishing, 2013.
2. Shashank Tiwari “Professional NoSQL”, 1st edition, Wiley India Pvt Ltd, 2015.
3. Provider Name: MongoDB University

Course Name: MongoDB for Developers

University: MongoDB University

Instructor: Andrew Erlichson

Course Link: <https://university.mongodb.com/courses/M101P/about>

4. Provider Name: edX.org

Course Name: Introduction to MongoDB using the MEAN Stack

University: MongoDB University

Instructor: Valeri Karpov

Course Link: <https://www.edx.org/course/introduction-mongodb-using-mean-stack-mongodbx-m101x-0>

Course Title : NOSQL DATABASE (P)	Course Code : 63P
Semester : VI	Course Group : DSC-XI
Teaching Scheme in Hrs (L:T:P): 0:0:4	Credits : 2 Credits
Map Code: M (PRACTICAL-APPLICATION)	Total Contact Hours : 60
CIA : 40 Marks	SEE # : 60 Marks
Programme : B.Sc.(IT)/BCA/B.Sc(CS)	# - Semester End Exam

LIST OF EXPERIMENTS

1. **Use Command:** Use Command-(Create a Student Database in MongoDB using “use” Command, Create an Employee Database in MongoDB using “use” Command).
2. Student Database and Employee Database-(Create a collection for student Database in MongoDB. (Consider the “db.collectionName” pattern).
3. Create a collection for Employee Database in MongoDB. (Consider the “db.collectionName” pattern).
4. **Run a Server:** Run a Server-(Run a server with the following instruction.-install mongodb : Open command prompt as administrator, Navigate to MongoDB root, Create a folder “data”, Run “mongod –dbpath ../data” Show command).
5. Show Command-(Use show command and list all database in the MongoDB, Use show command and list all collection in the Mongo shell).
6. Customize Prompt-(Customize prompt to Display database and hostname in the Mongo Shell).
7. **CRUD operation:** Find() method-(Use find () method for the following statement: Select all documents in an inventory collection, Select all documents in an inventory collection and “status is D”, Select documents in an inventory collection and apply “in” condition for status “A” and “D”, Select documents in an inventory collection and apply “and as well as or” condition).
8. Update() Command-(Use update() command for the following statement: Apply updateone command and update a single document for inventory collection (item=paper, size.uom=cm and status=p), Apply updatemany command and update a multiple document for inventory collection(qty is less than 50, size.uom=in and status=p)).
9. Delete Many and Remove command-(Use delete Many command and delete all documents from the inventory collection, Remove only one document that matches a “status is D” in

inventory collection).

10. **Text Search and Indexes:** Text Search-(Create text index for stores collection and fields are name=text , description=text, Use \$match and \$group operator and calculate the total views for articles that contains a word (cake) in article collection).
11. Indexes-(Use \$sort and \$meta operator and return results sorted by text search score for article Collection.
12. Create an ascending index on a single field and create an index on embedded document for record collection)
13. **Replication on mongodb:** Replica Set-(Create the replica set in the mongo shell and test the configuration)

REFERENCES:

1. Kristina chodorow “MongoDB, The definitive guide”, 2nd Edition, O’Reilly Publishing, 2013.
2. Shashank Tiwari “Professional NoSQL”, 1st edition, Wiley India Pvt Ltd, 2015.

3. Provider Name: MongoDB University

Course Name: MongoDB for Developers

University: MongoDB University

Instructor: Andrew Erlichson

Course Link: <https://university.mongodb.com/courses/M101P/about>

4. Provider Name: edX.org

Course Name: Introduction to MongoDB using the MEAN Stack

University: MongoDB University

Instructor: Valeri Karpov

Course Link: <https://www.edx.org/course/introduction-mongodb-using-mean-stack-mongodbx-m101x-0>

Course Title : RESPONSIVE WEB DESIGN(T)	Course Code : 63B
Semester : VI	Course Group : DSC - XII
Teaching Scheme in Hrs (L:T:P) : 5:1:0	Credits : 6 Credits
Map Code : D (THEORY – APPLICATION)	Total Contact Hours : 90
CIA : 25 Marks	SEE # : 75 Marks
Programme : BCA	# - Semester End Exam

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Understand the basic concepts of WEB and HTML structure	PSO1 & PSO 3	18	AP
CO2	Create Static Web page with List, Image, Links and Tables	PSO1 & PSO3	10	AP
CO3	Apply style to the static web page and Use CSS libraries such as Bootstrap to create responsive layouts	PSO1 & PSO3	8	AP
CO4	Implement JQuery selectors, variables and functions Manipulate web page content using jQuery	PSO1 & PSO3	18	AP
CO5	Use JavaScript data structures such as arrays and objects to define the data used in a web page	PSO1 & PSO3	18	AP
CO6	Use the Handlebars template library to convert data to HTML and add interactivity to templates using JavaScript event listeners.	PSO1 & PSO3	18	AP

UNIT – I**LECTURE HOURS: 18**

Html – Introduction: Web basics – (WWW, Client Side Scripting, Server Side Scripting, and Html). Set up the dev. environment – (Text file, Plain Editor, Text Wrangler, notepad++, Sublime Text, Example). Write some HTML- (Basic html structure, Example).**Nav Bar:** Build a nav bar- (White Space, Navigation, Nav bar, related examples).**Images and Lists:** Properly structured

documents- (Correct structure, Meta, Image tag, image as link List OL, UL). The World's first web page revisited-(Page View, Purpose of the page). **Responsive Web Design** : What does 'Responsive' mean?-(Definition, Advantages, Structure, Example).**Tutorial: Static Web Page:** Static Web page Development – (Html Structure).

UNIT – II

LECTURE HOURS: 18

Css – Introduction: Definition, Purpose Types of style sheets – (Inline, internal, External, Various Selectors, Examples). Fonts, Styles – (Graceful degradation, Structure of external sheets, color, style, size, Examples).**Links:** Customizing – (Need of customization, Pseudo selectors). **Layout Control** : Responsive layouts – (Text, color, body, indent). Media Queries- (What is a Media Query? - @media rule - Add a break point - Always Design for Mobile First - Another Breakpoint- Orientation: Portrait / Landscape).**Bootstrap:** Introduction – (Advantage – add bootstrap files – combined with CSS). Responsive grid – (Division- separate parts of the document - BS grid – device based grid).**Tutorial: Responsive Page:** Responsive Web page Development – (Bootstrap Elements)

UNIT – III

LECTURE HOURS: 18

jQuery – Introduction: Accessing Html tags, content – (Interactivity - purpose of JS).**jQuery Selectors:** Element, ID selector – (Syntax and examples). Class selector, separate function – (Syntax and examples).**jQuery In Action:** Responding to mouse clicks – (Accessing Html elements, properties , data). The Console-(Purpose, example). Manipulating HTML elements – (Way of manipulating, Script tag) .**Function:** User defined functions – (JavaScript function, calling statement, inline function).Anonymous function – (Advantages, Example).**Image Gallery Example:** Creating an Image gallery- (Step wise creation, Example).**Variables:** Using variables – (Syntax, Example). Variables in practice – (Syntax, Example).**If Statements:** Applying conditional statements – (Syntax, Example). Boolean variables - (Syntax, Example).**Tutorial: Ereader Example** : Creating an Image gallery – (Step wise creation, Example)

UNIT – IV

LECTURE HOURS: 18

Web Design Principles – Introduction: Module introduction, Usability and user experience – (Usability - User Experience (UX), Know your Audience). cognition based design, user centered design**Navigations:** Where can you go? – (Hierarchical, Local Global, Sitemaps Way of

visualizing of your websites).**Front End Web Development:** What is here? – (Box model, F View), Accessibility – (WAI, Accessibility and disability), Guidelines for design – (cognition based design, user centered design), Design examples – (Good and Bad Design examples, Dark Pattern).**Bootstrap:** Basic bootstrap setup – (Graceful degradation, Container class, bootstrap libraries).**Bootstrap In Action:** Where can you go?, Collapsible navbars – (Wrapping, active class, Example), Where am I? – (Breadcrumb trails), Responsive grid – (Column, BS code). Bootstrap themes – (Predefined themes to project).**Tutorial: Reader Example:** Dynamic Web Development – (Wireframes, nav bar)

UNIT – V

LECTURE HOURS: 18

Adding Content To Websites : Separating structure and content – (Identification of Structure and content), JavaScript objects – (Defining, Properties, member variables).**Javascript Templates With Handlebars :** Putting data into templates – (Initialize a template Expression, handlebar Example) , Displaying the same data with different templates, Use a Bootstrap modal-(Object- Template relationship , Example), Displaying the same data with different templates, Use a Bootstrap modal-(Modal – rendering templates- Example).**Javascript Arrays:** Storing objects in arrays and displaying them with a template – (Declaration, storing values, Nested Array) , Dynamically displaying single images from an array – (Template- Array, For each structure) , Implementing a search function – (Key press events, filters).**Data Structure For A Complete Image Gallery :** Writing the templates for the gallery – (Array – member variable , object, Apply Array to template), Switching displays in the gallery-(BS carousel, slide show).**Tutorial: Templates:** Image gallery – (Templates, JS Functionality).

TEXT BOOK:

T1. Ben Frain ,”Responsive Web Design with HTML5 and CSS3”, Packt Publishing, Edition 1, 2012.

REFERENCE BOOKS:

- R1. JakeSpurlock, “Bootstrap”, O'Reilly Publications, Edition 1, 2013
- R2. Crespo,“Responsive Web Design with JQuery”, Packt Publishing, Edition 1, 2012
- R3. <https://www.coursera.org/specializations/website-development>

Course Title : ANDROIDPROGRAMMING II (T)	Course Code : 63C
Semester : VI	Course Group : DSE-IV
Teaching Scheme in Hrs (L:T:P) : 4:0:0	Credits : 4 Credits
Map Code : E(THEORY-PROGRAMMING)	Total Contact Hours : 60
CIA : 25 Marks	SEE : 75 Marks
Programme : B.Sc.CS/ B.Sc IT/BCA	#-Semester End Exam

UNIT - I**LECTURE HOURS: 12**

How to work with styles: How to define a style(A style that overrides one property, A style that inherits a user defined style, A style that inherits a user defined style and overrides two properties , Another way to code the previous style , How to inherit multiple user defined styles) – How to apply a style(The reference chooser dialog for a style) - How to create a style sheet-(A styles.xml file in the res\values directory with four user defined styles)**How to work with themes:** How to modify a theme(A styles.xml file in the res\values directory with four user defined styles)-How to modify the text appearance for a theme(Some built in styles for controlling text appearance)-A summary of built in themes(Some built in themes)-How to apply themes(Theme.Holo.Light.Dialog,An AndroidManifest.xml file that uses a built-in theme) -**How to work with colors:-**How to define colors(A colors.xml file in the res\values directory)-How to apply colors (How to apply colors to a theme) **How to work with menus:**An introduction to menus(An activity with an options menu that has two items , The same options menu displayed from an action overflow icon)-How to define a menu(The file that contains the XML for the menu , The XML for the menu, some attributes of a menu item)-How to display options menu(CreateOptionsMenu method , inflate)-How to handle option menu events(The code that displays the menu, The code that handles the menu item events) -How to start a new activity(Code that starts a new activity, Code that uses menu items to start new activities)- **How to work with preferences:** An introduction to preferences(The settings activity , The dialog for the rounding item)-How to define preferences(The file that contains the xml for the preferences, The xml for the preferences, Some attributes that apply to all preference items, Some attributes that apply to List preference)-How to display preferences in an activity(The SettingsActivity class)-How to

display preferences in a fragment(The Settings Fragment class, The SettingsActivity class)-
 How to enable and disable preferences(A settings activity that uses dependencies , The xml
 for the preferences, The dependency attribute)- **Practical:** Use built in themes(Java and
 XML code)-Use styles(Java and XML code)-Modify a theme-Experiment with Menus and
 settings(Java and XML code)-Work with menus and settings(Java and XML code)-Work
 with Preferences(Java and XML code).

UNIT - II

LECTURE HOURS: 12

How to use single pane layouts for small screens: How to create the layout for a
 fragment(Start xml code for a fragment, use Graphical layout editor for a fragment)-How to
 create the class for a fragment(onCreate method and OnCreateView Method)-How to display
 a fragment in an activity(Add fragment element and use name attribute to specify fully
 qualified name for the class.)-How to create a preference fragment(Fragment needs to extend
 PreferenceFragment class).-How to display a preference fragment in an activity(Use
 addPreferenceFormResource method to add the Preferences defined in XML file.)- **How to
 use Multi-pane layouts for large screens:** -How to add multiple fragments to a layout.(Add
 two or more fragment elements to a layout file)-How to detect large screens(Create a values
 directory)-How to detect screen width(Create a values directory that uses smallest-width
 qualifier.)-How to control the soft keyboard(Delete requestFocus element from body of
 EditText element). **How to work with Threads:-**How thread work(Single sequential flow of
 control within a program)-How to execute asynchronous tasks(This task runs in a separate
 thread in background and does not to be synchronized.)-How to execute timed tasks(Use
 TimerTask and Timer classes to create a thread that execute at specific interval)-How to
 update UI thread(Use post method of a view object to execute any runnable object on UI
 thread)-How to download a file from Internet(Use Java and Android APIs to download a file
 by reading input from Internet and writing output to file system.)-How to parse an XML
 file(Use SAX to parse XML files.)- The RSSFeedhandler class(Extends DefaulterHandler
 which contains override ,First two statements define instance variables)-The RSSFeed
 class(Stores data for RSS feed which includes feed's title and publication date)-**How to work
 with Adapters:-**How to create the Layout for a list view(Stores data for RSS feed which
 includes feed's title and publication date)-How to use an adapter to display data in a list
 view(Use SimpleAdapter class to display data.)-How to handle events for an
 adapter(onItemClick method). **How to work with Intents:-**How to pass data between

activities(Explicit intent specify a component).-How to view a URL in a web browser(Implicit intent ,code passes ACTION_VIEW constant as first argument).-How to dial or call a phone number(ACTION_DIAL or ACTION_CALL constant).- **Practical:-** Create a new fragment(Java and XML Code)-Use Fragment manager(Java and XML Code)-Work with asynchronous tasks(Java and XML Code).

UNIT - III

LECTURE HOURS: 12

How to work with Application Object: How to define the Application object(Extend Application class and add instance variables and methods.)-How to register the Application object(To register ,open AndroidManifest ,XML file and edit application element.)-How to use Application Object(To register ,open AndroidManifest ,XML file and edit application element.). **How to work with services:-**The lifecycle of service()-How to create a service(Implement : the service, Intent,Ibinder classes).-How to register a service(Open manifest file and add service element at same indentation level).-How to start and stop a device(Start a service- calling startService method , provide menu Item that stops the service.)-How to use threads with services(Use - AsyncTask class).-How to test a service(Use - AsyncTask class).-How to view all services(Use - AsyncTask class).- **How to work with notification:-**How notification work(Appears as an icon in notification area at top screen, click)-How to create a pending intent(Passed to other app and execute, back stack sorted in order, task is cohesive unit contain multiple activities.)-How to create notification.(Use NotificationCompat Builder class)- **How to work with system services:-**How to display or remove notification(Use NotificationManager to display or remove a notification.)-How to check ,if a network is available(Use ConnectivityManager to check network available). **An introduction to databases:-** User interface for TaskList app(App provides personal and business,to display click on tabs, edit a task.click Delete tasks button to remove all completed tasks.)-An introduction to SQLite(Three data types (String,Int,Double)).-An introduction to task list database(Explanation - CREATE TABLE ,DROP TABLE).- **How to create a database class:-**How to define the constants for a database(Define constants for name and version. Tables names, column names, column numbers).-How to create or upgrade a database(Build- CREATE TABLE ,DROP TABLE,ALTER TABLE.)-How to open or close a database connection(Use methods – getReadableDatabase(), getWritableDatabase(), close())- **How to add public methods to database class:-** How to retrieve multiple rows from a table(Use WHERE clause).-How to retrieve a single row from a table(Works by getTask

Method).-How to get data from a cursor(Use getXxx method to get data).How to insert, update and delete rows(Use methods Insert (table,columns,values), update M(table, values, where, whereargs), Delete (table, where, whereargs)).- **How to test the database class and clear its data:-** How to test the database class(**Write** code to create an instance of DB class and use public methods. After building display in TextView widget)-How to clear test data from a device(Use settings app to delete the database for an app.).-How to use DDMS perspective to work with database files(Select directory and click “push a file”).-How to use the SQLite Database browser(Use – SQL DB browser and install SQLite DB browser. Create a new DB,open,view and execute SQL statements.). **Practical:-** Work with a service(Java and XML Code)-Use SQLite Database Browser(Java and XML Code)-Modify the database class for the Tasklist app(Java and XML Code).

UNIT - IV

LECTURE HOURS: 12

How to use tabs : How to add the TabManager class to project(Get TabManager jar file)-The Layout for an activity that display tabs(TabHost widget defines a container for tabs)-The class for an activity that displays tabs(TabManager object to add one or more tabs)-The class for a fragment that uses a custom adapter(Use Tabhost class to get tag for current tab).**How to use custom adapters:** A layout for a list view(ListView widget used to display list of tasks).-A class that extends the layout for a list view item.(Listview Widget contains complex widgets that extends layout)-A class for a customer adapter.(Extends BaseAdapter class) A class for a fragment that uses a custom adapter.(Use custom adapter to display appropriate data) **The Task List app:** The user interface(Two list :Personal & business).-The activity_task_list menu.(Add Task item , Delete item)-The TaskListActivity class()-The activity_add_edit and spinner_list layout.(Use spinner to select name ,user can view editable text to edit name and notes.)-The activity_add_edit menu(Defines menus ,display two items 1.save item 2.cancel item)-AddeditActivity class(Creates TaskListDB) **Practical:** Work with Tabs(Java and XML Code)- Work with a custom adapter(Java and XML Code)

UNIT - V

LECTURE HOURS: 12

An introduction to app widgets: A home screen that has app widget(App icons,App widgets,control widgets.create app widgets)-How to add app widgets to home screen(Follow Procedure for Android 3.0 and later,Prior to Android 3.0). **How to create app widgets:** How to create layout(Use Graphical layout layout editor).-How to modify the database class.(

Broadcast an action)-How to create the provider class(Extend AppwidgetProvider class).- How to configure an app widger(Add an info file in res\xml directory.).-How to register an app widget.(Appwidgetprovider extends broadcastreceiver class)-How to test an app widget(add app widget and test). **An introduction to distribution and monetization:** -How distribution works(Application marketplace or distribute directly to users via email or works).-How monetization works(Getting paid for an app,app billing provides in-app products.). **How to create a release build :** How to prepare an app for release(After testing and debugging ,release y final check up).-How to create the signed APK file(App must store in APK file signed with digital certificate). **How to distribute directly to the user:** How to distribute via a web site(Create mobile friendly web page).-How distribute via email(Attach APK file.)-**How to publish on Google play:-** How to set up a publisher account(Upload APK file and set up publisher account).-How to use the developer console to publish an app(Name of app,brief description of app,two screenshots of an app,an image of an app.).- How to view the listing for an app(Google play and search for an app).**Practical:-**Review and modify tasklist app widget(Java and XML code)- Add an app widget that displays a count of task(Java and XML code)-Install tip calculator app on your device(Java and XML code)-Publish an app on Google play.(Java and XML code)

TEXT BOOK:

T1. Training and Reference Murach's Android programming by Joel Murach , Mike Murach & Associates ,Inc.Chapters 7 , 8 , 9 , 10 , 11 , 13 , 14 , 16 , 17

REFERENCE BOOK:

- R1. Beginning Android Application Development by Wei-Meng Lee and Wiley Publishing, Inc. 2011
- R2. Android Programming Tutorials by Mark L. Murphy and CommonsWare, LLC., 2011
- R3. Android Apps for Absolute Beginners Edition:2 A Press Wallace Jackson(2013)

Course Title : ANDROID PROGRAMMING II (P)	Course Code : 63Q
Semester: VI	Course Group : DSE-IV
Teaching Scheme in Hrs (L:T:P) : 0:0:4	Credits : 2 Credits
Map Code : M(PRACTICAL-APPLICATION)	Total Contact Hours: 60
CIA : 40 Marks	SEE : 60 Marks
Programme: B.Sc CS/ B.Sc IT/BCA	#- Semester End Exam

LIST OF PROGRAMMS

1. To create a android application for Arithmetic Operation using android studio.
2. To create an android application for Styles using switch control
3. To create an android application for shared preference
4. To create an android application for Passing values between activity using Intent
5. To create an android application for student database
6. To create an android application for working with services
7. Create an android application for displaying toast message.

TEXT BOOK:

T1. Training and Reference Murach's Android programming by Joel Murach , Mike Murach & Associates ,Inc.Chapters 7 , 8 , 9 , 10 , 11 , 13 , 14 , 16 , 17

REFERENCE BOOK:

- R1. Beginning Android Application Development by Wei-Meng Lee and Wiley Publishing, Inc. 2011
- R2. Android Programming Tutorials by Mark L. Murphy and CommonsWare, LLC., 2011
- R3. Android Apps for Absolute Beginners Edition:2 A Press Wallace Jackson(2013)

Course Title : PROJECT & VIVA VOCE	Course Code : 63R
Semester : VI	Course Group : DSE-V
Teaching Scheme in Hrs (L:T:P) : 6:0:0	Credits : 6 Credits
Map Code: I(PROJECT)	Total Contact Hours : 90
CIA : 40 Marks	SEE # : 60 Marks
Programme: BSc.CS/BCA/BSc IT	# - Semester End Exam

During VI semester the students undergoes an individual project and each student will be assigned a domain. The project will be submitted at the end of the VI semester and will be jointly evaluated by the internal and external examiner.