B. Sc. Computer Science

Syllabus

(For students admitted from 2021-2022 onwards)

Department of Computer Science
Bishop Heber College (Autonomous)

Nationally Re-accredited at the 'A' by NAAC with a CGPA of 3.58 out of 4
Recognized by UGC as " College of Excellence"
Tiruchirappalli 620017

VISION

The Department of Computer Science is driven to provide excellent educational opportunities that accomplished the needs of our students, and empower them as an active technocrat in the top – notch IT industry and nation building.

MISSION

- Facilitating the quality technical education through enriched curriculum to solve the real-world problems.
- Creating the knowledge of innovative and sustainable research areas of computational science to build technological advanced society/nation.
- Educating the professional ethics, attitude, human values and career building skills for their professional and personal life.

PROGRAMMING OUTCOMES (POs)

Upon completion of BSc Computer Science degree, under-graduates will possess the following

computer science skills and abilities.

PO1: Acquiring the knowledge through a set of required courses covering essential areas in computing and a set of technical electives enabling students to deepen their knowledge in chosen areas of computational sciences.

PO2: Imbibing a strong computational oriented technical basis with a flexible interdisciplinary component and an emphasis on communication skills.

PO3: Applying the analytical skills to assess the problem and identify its solution using appropriate development of applications.

PO4: Design the applications using the programming skills of latest computing languages for societal needs and business use-cases.

PO5: Proposing original ideas and solutions, culminating into a modern, easy to use tool, by a larger section of the society with longevity.

PO6: Rendering eminent employability platform as a Up Grad professionals in a significant and indigenous sectors.

PO7: Perform effectively as an individual and as a member or a leader in a multidisciplinary setting to accomplish a goal.

PO8: Apply ethical principles and commit to professional ethics and responsibilities and norms of the scientific practice.

PO9: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

PROGRAMME SPECIFIC OUTCOMES (PSOs)

On the completion of the course the graduate will be able to

PSO1: Demonstrate the analytical and technical skills in the development of applications development using the knowledge of programming languages, operating systems, database management systems, computer graphics and software engineering.

PSO2: Integrate with the computer science community by the ability to function effectively on teams to accomplish shared computing design, evaluation, or implementation goals and also develop an entrepreneurship spirit.

PSO3: Exhibit the skills for getting the higher studies and employment in the computing and inter-disciplines for the development of professional and societal needs.

PSO4: Apply the technical skills for the development of rural areas and technically illiterate areas by providing the solutions of their problems to build the digitally empowered nation.

Programme Articulation Matrix - B.Sc., Computer Science

Sem	Course Code	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2	PSO3	PSO4
I	U18TM1L1				H	H	H	H	H	H				
	U21EG1L1				H	H	H	H	H	H				
	U21CS101	H	H	H	H	H				H	H	H	H	H
	U21CS1P1	H	H	H	H	H				H	H	H	H	H
	U20MAZ11	M	M	M	H	H				M	M	M	M	H
	U16EST11				H	H	H	H	H	H				
	U15VL1:1/ U15VL1:2				H	H	H	H	H	H				
II	U18TM2L2				H	H	H	H	H	H				
	U21EG2L2				H	H	H	H	H	H				
	U21CS202	H	H	H	H	H				H	H	H	H	H
	U21CS2P2	H	H	H	H	H				H	H	H	H	H
	U20MAZ22	M	M	M	H	H				M	M	M	M	H
	U20MAZ23	M	M	M	H	H				M	M	M	M	H
	U21CSPS1	H	H	H	H	H			H	H	M	H	H	H
III	U15TM3L3				H	H	H	H	H	H				
	U21EG3L3				H	H	H	H	H	H				
	U21CS303	H	H	H	H	H				H	H	H	H	H
	U21CS3P3	H	H	H	H	H				H	H	H	H	H
	U21PHZ34		H	H	H				H	H	M	M	M	H
	U21PHZP1		H	H	H				H	H	M	M	M	H
	U21CS3E1	H	H	H	H	H			H	H	L	L	L	M
VI	U15TM4L4				H	H	H	H	H	H				
	U21EG4L4				H	H	H	H	H	H				
	U21CS404	H	H	H	H	H				H	H	H	H	H
	U21CS4P4	H	H	H	H	H				H	H	H	H	H
	U21PHZ45		H	H	H				H	H	M	M	M	H
	U21PHZP1		H	H	H				H	H	M	M	M	H
	U21CS4E2	H	H	H	H	H			H	H	L	L	L	M
	U16LFS41				H	H	H	H	H	H				
	U16ETA41				H	H	H	H	H	H				
V	U21CS505	H	H	H	H	H				H	H	H	H	H
	U21CS506	M	M	M	H	H					H	H	H	H
	U21CS507	L	M	M	M	H	H			H	M	M	M	H
	U21CS5P5	H	H	H	H	H					H	H	H	H

	U21CS5:1	L	H	L	H	H	H	H		H	M	M	H	H
	U21CS5:2	H	H	H	H	M	M	H	M	H	H	H	H	H
	U21CS5:3				H	H	H	H	H	H		H	H	H
VI	U21CS608	H	H	H	H	H			M		H	H	H	H
	U21CS6:1	M	H	H	M	H			H		M	M	H	H
	U21CS6P6	M	H	M	M	H			M		M	M	M	M
	U21CS6:4	M	H	H	M	H			H		M	M	H	H
	U21CS6PJ	H	H	H	H	M	M	H	M	H	H	H	H	H
	U16GST61				H	H	H	H	H	H				

Programme Structure - B.Sc., Computer Science
(Applicable to Candidates admitted From the Academic Year (2021-2022) onwards

Sem	Part	Course	Course Title	Course Code	Hours / Week	Credits	Marks		
							CIA	ESE	Total
I	I	Tamil I	செய்யுள், இலக்கியவரலாறு,உரைநடை,மொழிப்பயிற்சியும் படைப்பாக்கமும்	U18TM1L1	6	3	25	75	100
	II	English I	Literature and Language: Prose and Short Stories	U21EG1L1	6	3	40	60	100
	III	Core I	Programming Principles and Data Structures using C	U21CS101	6	5	25	75	100
		Core Practical I	Programming Principles and Data Structures using C Lab	U21CS1P1	3	2	40	60	100
		Allied I	Operation Research	U20MAZ11	5	4	25	75	100
		Env. Studies	Environmental Studies	U16EST11	2	2	25	75	100
	IV	Val.Edu.	Value Education (VLO - RI / MI)	U15VL1:1/ U15VL1:2	2	2	25	75	100
			,,,,,,,,,,,,,,,						
II	I	TamilII	செய்யுள், இலக்கியவரலாறு,சிறுகதைத்தரட்டு,மொழிப்பயிற்சிரு	U18TM2L2	6	3	25	75	100
	II	English II	Literature and Language: Poetry and Shakespeare	U21EG2L2	6	3	40	60	100
	III	Core II	Python Programming	U21CS202	5	5	25	75	100
		Core Practical II	Python Programming Lab	U21CS2P2	3	2	40	60	100
		Allied II	Numerical Methods	U20MAZ22	4	4	25	75	100
		Allied III	Probability and Statistics	U20MAZ23	4	4	25	75	100
	IV	SBEC I	Business Communication and Collaboration Tools	U21CSPS1	2	2	40	60	100
III	I	TamilIII	செய்யுள்,காப்பியங்கள், இலக்கியவரலாறு,நாவல் மொழிப்பயிற்சி	U18TM3L3	6	3	25	75	100
	II	English III	English for Competitive Examinations	U21EG3L3	6	3	40	60	100
	III	Core III	Object Oriented Programming Concepts	U21CS303	6	5	25	75	100
		Core Practical III	Object Oriented Programming Concepts Lab	U21CS3P3	3	2	40	60	100
		Applied I	Electricity, Magnetism and Electromagnetism	U21PHZ34	4	3	25	75	100
		Applied Practical I	Applied Physics Practicals	U21PHZP1	3	---	---	---	---
	IV	NMEC I	E – Commerce	U21CS3E1	2	2	25	75	100

Sem	Part	Course	Course Title	Course Code	Hours / Week	Credits	Marks		
							CIA	ESE	Total
IV	I	Tamil IV	செய்யுள்,நாடகம், இலக்கியவரலாறு,மொழிப்பயிற்சி	U18TM4L4	5	3	25	75	100
	II	English IV	English through Literature	U21EG4L4	5	3	40	60	100
	III	Core IV	Design and Implementation of Database	U21CS404	6	5	25	75	100
		Core Practical IV	Design and Implementation of Database Lab	U21CS4P4	3	2	40	60	100
		Applied II	Solid State Devices and Microprocessor	U21PHZ45	4	4	25	75	100
		Applied Practic II	Applied Physics Practicals	U21PHZP1	3	3	40	60	100
	IV	NMEC II	Digital Marketing	U21CS4E2	2	2	25	75	100
		Soft Skills	Life Skills	U16LFS41	2	1	---	---	100
	V	Extension Activities	NSS, NCC, Rotract, Lioclub, etc…,	U16ETA41	---	1	---	---	---
V	III	Core V	Web Programming	U21CS505	5	5	25	75	100
		Core VI	Principles of Operating Systems	U21CS506	5	5	25	75	100
		Core VII	Foundation of Computer Graphics	U21CS507	5	5	25	75	100
		Core Practical V	Web Programming Lab	U21CS5P5	6	4	40	60	100
		Elective I	1a) Software Engineering	U21CS5:1	5	5	25	75	100
			1b) Internet of Things	U21CS5:2					
			1c) Cloud Computing	U21CS5:3					
	IV	SBEC II	Mini Project	U21CSPS2	2	2	40	60	100
		SBEC III	Technical Communication for Computer Technocrats	U21CSPS3	2	2	40	60	100
VI	III	Core VIII	Computer Networking	U21CS608	6	5	25	75	100
		Elective II	2a) Machine Learning	U21CS6:1	6	5	25	75	100
			2b) Web Applications Development	U21CS6:2					
			2c) XML Basics	U21CS6:3					
		Elective III	3a) Business Analytics	U21CS6:4	6	5	25	75	100
			3b) Foundations of Enterprise Computing	U21CS6:5					
			3c) Mobile Application Development	U21CS6:6					
		Core Practical VI	Digital Electronics and Microprocessors Lab	U21CS6P6	6	5	40	60	100
		Core / Project	Project	U21CS6PJ	6	5	---	---	100
	V	Gender Studies	Gender Studies	U16GST61	---	1	---	---	100
Total Credits						**140**			

CORE I: PROGRAMMING PRINCIPLES AND ABSTRACTIONS USING C

SEMESTER: I **CODE: U21CS101**
CREDITS: 5 **HOURS /WEEK: 6**

---

1. COURSE OUTCOMES

CO #	Course Outcomes	Level	Unit
CO1	Examine the basic concepts of C programming and use them in creating simple applications	K4	1
CO2	Apply the looping concepts and different types of arrays for solving computing problems	K3	2
CO3	Develop applications using the concept of pointers, declaration, initialization, operations on pointers and its usage	K6	3
CO4	Create a file in order to read, write, display and append data using file functions	K6	4
CO5	Select the appropriate data structures based on the requirements of the problem.	K5	5
CO6	Develop the Programs for Data Structures with real world use-cases	K6	5

2. A. SYLLABUS

Unit I:Language Basics, Types, Expressions and Operators

Language Basics: Characteristics of C - Structure of C Programs - Source Files - Comments - Character Sets - Identifiers - How the C Compiler Works- Types: Typology - Integer Types - Floating-Point Types - Complex Floating-Point Types -Enumerated Types - Type void – Literals: Integer Constants -Floating-Point Constants - Character Constants - String Literals - Type Conversions: Conversion of Arithmetic Types & Nonarithmetic Types - Expressions and Operators: Evaluation of Expression- Operators Constant Expressions

Unit II: Statements, Functions and Arrays

Statements: Expression Statements - Block Statements - Loops - Selection Statements - Unconditional Jumps. Functions: Function Definitions - Function Declarations – Execution - Pointers as Arguments and Return Values - Inline Functions - Non-Returning Functions - Recursive Functions - Variable Numbers of Arguments. Built-in Functions: Input and Output- Mathematical Functions - Character Classification and Conversion - String Processing - Multibyte Characters - Converting Between Numbers and Strings- Searching and Sorting. Arrays: Defining Arrays - Accessing Array Elements- Initializing Arrays - Strings - Multidimensional Arrays Arrays as Arguments of Functions

Unit III: Pointers, Structures and Union, Files

Pointers: Declaring Pointers - Operations with Pointers - Pointers and Type Qualifiers - Pointers to Arrays and Arrays of Pointers - Pointers to Functions. - Structures, Unions, and Bit-Fields: Structures - Unions Anonymous Structures and Unions - Bit-Fields. Input and Output: Streams - Files - Opening and Closing Files - Reading and Writing - Random File Access

Unit IV: Linear Data Structures

Introduction: Data Structures - Abstract Data Types (ADTs). Stacks: Stack - Stack ADT- Applications – Implementation - Comparison of Implementations – Queue: Queue ADT – Exceptions – Applications – Implementation - Linked Lists - Linked Lists ADT - Comparison of Linked Lists with Arrays & Dynamic Arrays - Singly Linked Lists - Doubly Linked Lists - Circular Linked Lists

Unit V: Non-Linear Data Structures

Graphs: Introduction – Glossary - Applications of Graphs - Graph Representation- Graph Traversals - Topological Sort – Trees : Glossary - Binary Trees - Types of Binary Trees - Properties of Binary Trees - Binary Tree Traversals – Binary Search Trees.

B. TOPICS FOR SELF - STUDY

S. No	Topic	Web links
1	Embedded Software in C	https://www.youtube.com/watch?v=CYvJPra7Ebk
2	Art of Code	https://www.youtube.com/watch?v=6avJHaC3C2U
3	Google Coding	https://www.youtube.com/watch?v=qz9tKlF431k
4	Structures and Unions	https://www.youtube.com/watch?v=KL65a0TyeYo

C. TEXT BOOKS

1. Peter Prinz and Tony Crawford, "C in a Nutshell", 2nd Edition, Oreilly Publishers, 2016.
2. Narasimha Karumanchi, "Data Structures And Algorithms Made Easy", 5ed, CareerMonk Publications, 2017.

D. REFERENCES

1. Steve Oualline, ***"Practical C Programming",*** 3ed, Oreilly Publishers, 1997.
2. Reema Thareja, ***"Data Structures using C",*** Oxford University Press, 2015.

E. WEB LINKS

- https://www.udemy.com/course/advanced-c-programming-course/
- https://www.coursera.org/courses?query=c%20programming
- http://www.ictacademy.in/pages/Advanced-C-Programming.aspx

3. SPECIFIC LEARNING OUTCOMES (SLO)

Unit/ Section	Course Content	Learning outcomes	Blooms Taxonomy Highest Level of Transaction
I	**Language Basics, Types, Expressions and Operators**		
1.1	**Language Basics**: Characteristics of C - Structure of C Programs - Source Files	Explain the structure of C Explain the characteristics of C	K5 K5
1.2	Comments - Character Sets - Identifiers - How the C Compiler Works	Explain the Character Sets in C Construct the identifiers using the rules of C	K4 K6
1.3	**Types:** Typology - Integer Types - Floating-Point Types -	Explain the types in C Build the integer based variable using types	K4 K5
1.4	Complex Floating-Point Types - Enumerated Types - Type void –	Explain the complex and point type Build the enumerated types in C	K4 K6
1.5	**Literals**: Integer Constants - Floating-Point Constants - Character Constants - String Literals	Compare the features of various constants List the literals in C	K6 K2

1.6	**Type Conversions:** Conversion of Arithmetic Types & Nonarithmetic Types	Develop the program for type conversions	**K6**
1.7	**Expressions and Operators**: Evaluation of Expression- Operators Constant Expressions	List the operators in C Design the program for evaluation of expression	**K3** **K6**
II	**Statements, Functions and Arrays**		
2.1	**Statements:** Expression Statements - Block Statements	Develop the program using if statements	K6
2.2	Loops - Selection Statements - Unconditional Jumps.	Develop the program using loop statements	K6
2.3	**Functions:** Function Definitions - Function Declarations – Execution	Explain the mechanism of Functions in C Develop the program using Functions	K4 K6
2.4	Pointers as Arguments and Return Values	Develop the program using Pointers as Arguments and Return Values	K6
2.5	Inline Functions - Non-Returning Functions	Explain the inline functions	K5
2.6	Recursive Functions - Variable Numbers of Arguments.	Develop the program using recursive functions	K6
2.7	**Built-in Functions:** Input and Output- Mathematical Functions - Character Classification and Conversion - String Processing	Explain the built-in functions Develop the program using Input and Output- Mathematical Functions	K4 K6
2.8	Multibyte Characters - Converting Between Numbers and Strings- Searching and Sorting.	Develop the program using conversion of numbers and strings	K6
2.9	**Arrays:** Defining Arrays - Accessing Array Elements- Initializing Arrays – Strings	Explain the arrays Develop the program using arrays	K4 K6
2.10	Multidimensional Arrays - Arrays as Arguments of Functions	Develop the programs using multidimensional arrays	K6
III	**Pointers, Structures and Union, Files**		
3.1	**Pointers:** Declaring Pointers - Operations with Pointers - Pointers and Type Qualifiers.	Develop the programs using pointers Explain the pointer operations	K6 K4
3.2	Pointers to Arrays and Arrays of Pointers - Pointers to Functions	Develop the program for Pointers to Arrays	K6
3.3	**Structures, Unions, and Bit-Fields**: Structures – Unions	Develop the program for structures and unions	K6
3.4	Anonymous Structures and Unions - Bit-Fields.	Explain the concept of Bit-Fields	K4
3.5	**Input and Output**: Streams - Files - Opening and Closing Files - Reading and Writing	Develop the program for File Processing List the functions for file processing	K6 K4
3.6	Random File Access	Explain the Random File Access	K4
IV	**Linear Data Structures**		
4.1	Introduction: Data Structures - Abstract Data Types (ADTs).– -	Construct the ADTs using Arrays	K6

4.2	**Stacks:** Stack - Stack ADT-Applications – Implementation - Comparison of Implementations	Explain the stack in detail Develop the program for Stack operations	K4 K6
4.3	**Queue:** Queue ADT – Exceptions – Applications **–** Implementation -	Explain the queue in detail Develop the program for queue operations	K4 K6
4.4	**Linked Lists -** Linked Lists ADT **-** Comparison of Linked Lists with Arrays & Dynamic Arrays	Explain the linked list Compare the Linked Lists with Arrays & Dynamic Arrays	K4 K6
4.5	Singly Linked Lists **-** Doubly Linked Lists **-** Circular Linked Lists	Assess the functionality of different types of linked list	K6
V	**Non-Linear Data Structures**		
5.1	**Graphs:** Introduction – Glossary - Applications of Graphs - Graph Representation	Explain the Graphs and its applications Evaluate the graph representation	K4 K5
5.2	Graph Traversals **-** Topological Sort **–**	Develop the program for graph traversals	K6
5.3	**Trees** : Glossary **-** Binary Trees	Explain the tree and its types	K4
5.4	Types of Binary Trees **-** Properties of Binary Trees	Assess the properties of binary trees	K6
5.5	Binary Tree Traversals	Develop the program for binary tree traversals	K6
5.6	Binary Search Trees.	Develop the program for binary search trees	K6

4. MAPPING SCHEME (CO, PO, PSO)

L-Low **M-Moderate** **H-High**

U21CS101	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	H	-	M	M	L	-	M	M	-	H	-		M
CO2	H	-	M	M	M	-	-	-	-	H	-		M
CO3	H	-	M	M	M	-	M	M	-	H	-		M
CO4	H	-	M	M	M	-	-	-	-	H	-		H
CO5	H	M	M	M	M	-	-	-	-	H	M	M	H
CO6	H	H	H	H	H	M	-	-	-	H	M	M	H

5. COURSE ASSESSMENT METHODS

DIRECT:

1. Continuous Assessment Test: T1, T2 (Theory & Practical Components): Closed Book
2. Open Book Test.
3. Assignment, Group Discussion, Poster Presentation, Seminar, Quiz (written).
4. Pre-Semester & End Semester Theory Examination

INDIRECT:

1. Course end survey (Feedback)

Name of the Course Coordinator : Prof. V. Bhuvaneswari

PROGRAMMING PRINCIPLES AND ABSTRACTIONS USING C LAB

SEMESTER: I **CODE: U21CS1P1**
CREDITS : 2 **HOURS /WEEK: 3**

----

1. COURSE OUTCOMES

CO #	Course Outcomes	Level	Unit
1	To identify the data types and use them in simple data processing applications	K2	1
2	To develop the ability to analyse a problem and develop an algorithm to solve it	K4	2
3	To analyse and relate the concept of pointers and their usage	K5	3
4.	To apply the concept of data structures like tree, stack and queue in the programming context	K6	4-8
5.	To interpret the programming task involved for a given computational problem	K6	4-8
6.	To Develop the file operations based programmes for file processing	K6	9-10

2. SYLLABUS

EXP NO	CONTENT
1.	Starter application: Read and print name, Convert temperature from F to C and vice versa, Find biggest of 3 numbers, Compute area of shapes: rectangle, circle and cylinder.
2.	Sorting student marks: Sort the given marks of students in ascending order and descending order
3.	Arithmetic calculator: Develop menu-driven arithmetic calculator application using functions. Each function should receive values and return the calculated output.
4.	[Sorting employees]: Each employee record contains name and salary. Read name and salary of 10 employees into structure. Sort the records based on salary, if same then sort on name
5.	[Mark List Generation]: Read a file that contains details of 5 students. Each record contains student name, mark1, mark2 and mark3. Prepare a mark list that displays student name, marks and average mark of each student.
6.	Sorting a stack: Given a stack, sort it such that the top of the stack has the largest element. Read the size of the stack, followed by the numbers to be pushed to stack. The output should be the popped elements from the sorted stack.
7.	Write C programs for the following: a. Uses stack operations to convert infix expression into postfix expression b. Uses stack operations for evaluating the postfix expression
8.	Car washing queue: A car washing garage has 2 gates one to enter another to exit and can wash 3 cars at a time. Cars can enter the garage in one gate and exit in another gate. For this application, ask user which operation one wants such as '1-insert car', '2-remove car', '3-show queue' or '4-quit'. Perform the following sequence of operations: 1 1; 1 2; 1 3; 2; 3; 1 4; 3; 2; 1 5; 1 6; 2; 3; 4
9.	Intersection of two linked lists: Given two linked lists, your task is to complete the function findIntersection(), that returns the intersection (ie., common elements) of two linked lists.

	The function takes 2 arguments, reference pointer to the head of the first linked list (head1) and reference pointer to the head of the second linked list (head2). The function should return reference pointer to the head of the new list that is formed by the intersection of the two the lists.
10.	Write C functions for manipulating a doubly linked list as follows: Add a node after p , Add a node before p , Delete node p and Traverse the list
11.	[Detecting cycles in a graph]: Given a directed graph, your task is to detect if there is a cycle in the graph or not. Read two integers 'N' and 'M' which denotes the no of vertices and no of edges respectively. Read 'M' pairs u and v denoting that there is an uni-directed edge from u to v. Print if there is a cycle or not.
12.	Construct a binary tree and binary search tree. Find the Traversals of Binary tree and binary search tree and print the elements of both.

3. SPECIFIC LEARNING OUTCOMES (SLO)

EXP NO	CONTENT	Blooms Taxonomy Highest Level of Transaction
1	Starter application: Read and print name, Convert temperature from F to C and vice versa, Find biggest of 3 numbers, Compute area of shapes: rectangle, circle and cylinder.	K1
2	Sorting student marks: Sort the given marks of students in ascending order and descending order	K3
3	Arithmetic calculator: Develop menu-driven arithmetic calculator application using functions. Each function should receive values and return the calculated output.	K2
4	rting employees]: Each employee record contains name and salary. Read name and salary of 10 employees into structure. Sort the records based on salary, if same then sort on name	K3
5	[Mark List Generation]: Read a file that contains details of 5 students. Each record contains student name, mark1, mark2 and mark3. Prepare a mark list that displays student name, marks and average mark of each student.	K3
6	Sorting a stack: Given a stack, sort it such that the top of the stack has the largest element. Read the size of the stack, followed by the numbers to be pushed to stack. The output should be the popped elements from the sorted stack.	K3
7	Write C programs for the following: Uses stack operations to convert infix expression into postfix expression Uses stack operations for evaluating the postfix expression	K3

8	Car washing queue: A car washing garage has 2 gates one to enter another to exit and can wash 3 cars at a time. Cars can enter the garage in one gate and exit in another gate. For this application, ask user which operation one wants such as '1-insert car', '2-remove car', '3-show queue' or '4-quit'. Perform the following sequence of operations: 1 1; 1 2; 1 3; 2; 3; 1 4; 3; 2; 1 5; 1 6; 2; 3; 4	K4
9	Intersection of two linked lists: Given two linked lists, your task is to complete the function findIntersection(), that returns the intersection (ie., common elements) of two linked lists. The function takes 2 arguments, reference pointer to the head of the first linked list (head1) and reference pointer to the head of the second linked list (head2). The function should return reference pointer to the head of the new list that is formed by the intersection of the two the lists.	K5
10	Write C functions for manipulating a doubly linked list as follows: Add a node after p , Add a node before p , Delete node p and Traverse the list	K6
11.	[Detecting cycles in a graph]: Given a directed graph, your task is to detect if there is a cycle in the graph or not. Read two integers 'N' and 'M' which denotes the no of vertices and no of edges respectively. Read 'M' pairs u and v denoting that there is an undirected edge from u to v. Print if there is a cycle or not.	
12.	Construct a binary tree and binary search tree. Find the Traversals of Binary tree and binary search tree and print the elements of both.	

4. MAPPING SCHEME (CO, PO, PSO)

L-Low M-Moderate H- High

U21CS1P1	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2	PSO3	PSO4
CO1	H	-	-	M	M	M	-	-	-	-	-	-	-
CO2	H	M	-	M	M	M	-	-	-	M	-	M	-
CO3	H	-	-	M	M	M	-	-	-	-	-	-	-
CO4	H	-	-	M	M	M	-	-	-	M	-	M	-
CO5	H	M	L	M	M	M	-	-	-	M	-	M	-
CO6	H	M	L	M	M	M	-	-	-	M	-	M	-

5. COURSE ASSESSMENT METHODS

DIRECT:

1. Continuous Assessment Test: T1, T2 (Practical Components): Closed Book
2. Cooperative Learning Report, Assignment, Group Discussion, project Report, Field Visit Report, Seminar.
3. Pre/Post Test, Viva, Report for each Exercise.
4. Lab Model Examination & End Semester Practical Examination

INDIRECT:

1. Course end survey (Feedback)

Name of the Course Coordinator : Prof. V. Bhuvaneswari

PYTHON PROGRAMMING

SEMESTER: II **CODE : U21CS202**
CREDITS : 5 **HOURS /WEEK: 5**

----

1. COURSE OUTCOMES

CO #	Course Outcomes	Level	Unit
CO1	Describe the core syntax and semantics of Python programming language.	K3	1
CO2	Discover the need for working with the strings and functions.	K5	2
CO3	Illustrate the process of structuring the data using lists, dictionaries, tuples and built-in functions to navigate the file system.	K5	3
CO4	Infer the Object-oriented Programming concepts in Python.	K4	4
CO5	Indicate the use of regular expressions andbuild a real time application using the python concepts	K6	5
CO6	Design the basic web applications using Python Internet Programming	K6	5

2. A. SYLLABUS

Unit I: Python Basics and Functions **6**

Variables – Operators – Statements – Getting Inputs – Boolean conditions – Alternative, chained and nested conditions – Catching exceptions – Function calls – Built-in functions – Type conversion functions and math functions – Creating new functions, parameters and arguments – Need for functions.

Unit II: Loops and Strings **6**

While statement – Infinite loops – Continue statement – For loops – Counting and summing loops – Maximum and minimum loops – Traversal through strings – String slice – Looping and counting in strings – The in operator – String comparison – String methods – parsing strings – Format operator.

Unit III: Files and Lists **6**

Opening files – Text files – Reading files – Searching through files – Selecting files names from user – Writing files – Traversing list – List operations – List slice – List methods – Deleting elements – Built-in list functions – Objects, value and aliasing – List arguments.

Unit IV: Dictionaries, Tuples and OOP **6**

Dictionaries – Files and dictionaries – Looping and dictionaries – Advanced text processing – Tuples – Comparing tuples – Tuple assignments – Dictionaries and tuples – Tuples as keys in dictionaries – Creating objects – Encapsulation – Classes as types – Object lifecycle – Instances – Inheritance.

Unit V: Internet Programming **6**

Regular expressions – Character matching – Extracting data – Escape character – Designing simple web browser using sockets – Retrieving images using HTTP – Retrieving web pages using urllib – Reading binary files using urllib.

B. TOPICS FOR SELF - STUDY

S.No.	Topics	Web Links
1	Speech Recognition in Python	https://www.youtube.com/watch?v=qz9tKlF431k
2	Algorithmic Trading	https://www.youtube.com/watch?v=SEQbb8w7VTw
3	Java Vs Python	https://www.youtube.com/watch?v=s3Ejdx6cIho
4	Files and Dictionaries	https://www.youtube.com/watch?v=mcWlhxp4dws&list=PLGYpb6ADpoKcWjgkktEHLpU1mGMb0ZTmj

C. TEXT BOOK

1. Charles R. Severance, Python for Everybody: "***Exploring data using Python 3"***, Schroff Publishers, 1ed, 2017, ISBN 978-9352136278.

D. REFERENCE BOOK

1. Allen Downey, Think Python: "***How to think like a computer scientist"***, Schroff / O'Reilly Publishers, 2ed, 2016, ISBN 978-9352134755.

E. WEB LINKS

- https://www.udemy.com/course/learn-advanced-python-programming/
- https://realpython.com/tutorials/advanced/
- https://python-course.eu/advanced-python/

3. SPECIFIC LEARNING OUTCOMES (SLO)

UNIT	CONTENT	LEARNING OUTCOMES	Blooms Taxonomy Highest Level of Transaction
I	Python Basics and Functions		
1.1	Variables, Operators, Statements, Getting Inputs	Interpret basics of programming Concepts.	K2
1.2	Boolean Conditions	Indentify the Python language control statements	K2
1.3	Alternative, Chained and nested conditions	Make out the python Nested Conditions	K2
1.4	Catching Exceptions	Apply and analysis the catching exceptions	K2
1.5	Function Calls	Illustrate the structure of python functions.	K3
1.6	Built-in Functions	Interpret the predefined library functions.	K3
1.7	Type Conversion functions and math functions	Infer the method of type conversion and functionalities of math functions.	K3
1.8	Creating new functions, parameters and arguments	Make out the creation of new python functions	K3

1.9	Need for Functions	Infer the need for functions.	K3
II	**Loops and Strings**		
2.1	While Statement, Infinite Loops, Continue Statement	Illustrate the Python language syntax including control statements.	K2
2.2	For loops	Illustrate the Python language loops.	K2
2.3	Loop patterns	inspect the loop concept and evaluate loop patterns.	K4
2.4	Traversal through strings, String slice	Illustrate the functionalities of String	K3
2.5	Looping and Counting in Strings	Interpret and evaluate the functionalities of strings	K3
2.6	The in operator, String comparison, String methods, Parsing Strings, Format operator	List out and evaluate the set of String functions	K4
III	**Files and Lists**		
3.1	Opening files, Text files and Reading	Determine the csv file using DataFrame Illustrate and evaluate the functions for reading tabular data as a DataFrame object	K5 K4
3.2	Search through file	Illustrate the easiest ways to store data efficiently in binary format.	K4
3.3	Selecting and writing file	Illustrate and apply the python APIs for connecting python with web	K4
3.4	Traversing List, List operations	Explain how to connect data base with python. Illustrate and evaluate the data and store it in database	K5 K4
3.5	Built in list functions	Illustrate and evaluate the Data contained in pandas objects can be combined together	K3
3.6	List arguments	Create and apply the functionalities for rearranging tabular data	K4
IV	**Dictionaries, Tuples and OOP**		
4.1	Files and Dictionaries	Apply and evaluate the data in dictionaries	K4
4.2	Looping and Dictionaries	Apply and evaluate the data in dictionaries using looping	K4
4.3	Advanced text processing	Apply and evaluate the data in advanced functionalities.	K4
4.4	Tuples, Comparison, assignments	Create and evaluate the data using tuples	K4
4.5	Dictionaries and tuples, Tuples as keys in Dictionaries	Illustrate and analysis the tuples as keys in dictionaries	K4
4.6	Creating Objects, Encapsulation	Illustrate the oops concepts	K3
	Classes as types, Object life cycle	Interpret the class types and object life cycle	K3
	Instances, Inheritance	Illustrate the instances and inheritance	K3
V	**Internet Programming**		
5.1	Regular Expressions	Illustrate and evaluate the regular expressions	K5
5.2	Character matching, Extracting Data, Escape Character	Illustrate and evaluate regular expression using escape character and extracting and matching data.	K5
5.3	Designing simple web browser using Sockets	Illustrate evaluate and create web browser using sockets	K5
5.4	Retrieving images using HTTP	Illustrate and evaluate images using HTTTP	K5
5.5	Retrieving web pages using urllib	Illustrate and evaluate the web pages using urllib	K5
5.6	Reading binary files using Urllib	Interpret the binary files using urllib Build the real time application using python library	K5,K6

MAPPING SCHEME (CO, PO, PSO)

L-Low **M-Moderate** **H-High**

U21CS202	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2	PSO3	PSO4
CO1	H	-	M	M	L	-	M	M	-	H	-		M
CO2	H	-	M	M	M	-	-	-	-	H	-		M
CO3	H	-	M	M	M	-	M	M	-	H	-		M
CO4	H	-	M	M	M	-	-	-	-	H	-		H
CO5	H	M	M	M	M	-	-	-	-	H	M	M	H
CO6	H	H	H	H	H	M	-	-	-	H	M	M	H

5. COURSE ASSESSMENT METHODS

DIRECT:

1. Continuous Assessment Test: T1, T2 (Theory & Practical Components): Closed Book
2. Open Book Test.
3. Learning Report, Assignment, Group Presentation, Group Discussion, project Report, Field Visit Report, Poster Presentation, Seminar, Quiz (written).
4. Pre-Semester & End Semester Theory Examination

INDIRECT:

1.Course end survey (Feedback).

Name of the Course Coordinator: Prof. S. Maheswari

PYTHON PROGRAMMING LAB

SEMESTER: II **CODE: U21CS2P2**
CREDITS : 2 **HOURS /WEEK: 3**

----

1. COURSE OUTCOMES

CO #	Course Outcomes	Level	Exercise No
CO1	Develop the python programming for data analysis using basic operations	K5	1-3
CO2	Design the python for date time module methods and operators	K5	4
CO3	Develop the python programming for solving the data analysis problems (sorting, removing duplicates)	K6	5-10
CO4	Develop the python programming to test for data analysis problems using list, tuples and dictionary.	K6	11-12
CO5	Design the applications for processing the text files	K6	13-14
CO6	Construct the database file CSV file and analysis that data.	K6	15

2. SYLLABUS

SIMPLE APPLICATIONS 15

1). Practice using the Python interpreter as a calculator:
A). The volume of a sphere with radius r is $4/3\pi r^3$. What is the volume of a sphere with radius 5?
B). Suppose, cover price of a book is INR 24.95, but a bookstore offers a 40% discount. Shipping costs INR 3.00 for the first copy and 75 Paise for each additional copy. What is the total wholesale cost for 60 copies?
C). If I leave my house at 6:52 am and run 1 km at an easy pace (8:15 per km), then 3 km at tempo (7:12 per km) and 1 km at easy pace again, what time do I get home for breakfast?
2). Develop an application to prompt for a score between 0.0 and 1.0. If the score is out of range, print an error message. If the score is between 0.0 and 1.0, print a grade using the following table:

Score	Grade	Score	Grade	Score	Grade	Score	Grade	Score	Grade
>= 0.9	A	>= 0.8	B	>= 0.7	C	>= 0.6	D	< 0.6	F

3). Develop an application that repeatedly reads numbers until the user enters “done”. Once “done” is entered, print out the total, count, and average of the numbers. If the user enters anything other than a number, detect their mistake using try and except and print an error message and skip to the next number.
4). The **datetime** module provides time objects that are similar to the Time objects, but they provide a rich set of methods and operators. Write a program that takes a birthday as input and prints the user’s age and the number of days, hours, minutes and seconds until their next birthday.
5). Given a list of strings, return the count of the number of strings where the string length is 2 or more and the first and last chars of the string are the same.
6). Develop a function front_x(). Given a list of strings, return a list with the strings in sorted order, except group all the strings that begin with 'x' first. Eg. ['mix', 'xyz', 'apple', 'xanadu', 'aardvark'] yields ['xanadu', 'xyz', 'aardvark', 'apple', 'mix']. Hint: this can be done by making 2 lists and sorting each of them before combining them.
7). Develop a function sort_last(). Given a list of non-empty tuples, return a list sorted in increasing order by the last element in each tuple. E.g. [(1, 7), (1, 3), (3, 4, 5), (2, 2)] yields [(2, 2), (1, 3), (3, 4, 5), (1, 7)]. Hint: use a custom key= function to extract the last element form each tuple.
8). Develop a function remove_adjacent(). Given a list of numbers, return a list where all adjacent same elements have been reduced to a single element, so [1, 2, 2, 3] returns [1, 2, 3]. List [2, 2, 3, 3, 3] returns [2, 3]. List [ ] returns [ ]. You may create a new list or modify the passed in list.
9). Write a function verbing(). Given a string, if its length is at least 3, add 'ing' to its end. Unless it already ends in 'ing', in which case add 'ly' instead. If the string length is less than 3, leave it unchanged. Return the resulting string. So ‘hail’ yields: hailing; ‘swimming’ yields: swimmingly; ‘do’ yields: do.

10). Develop a function not_bad(). Given a string, find the first appearance of the substring 'not' and 'bad'. If the 'bad' follows the 'not', replace the whole 'not'...'bad' substring with 'good'. Return the resulting string. So 'This dinner is not that bad!' yields: This dinner is good!

II. ADVANCED APPLICATIONS **15**

11) Develop an application that prompts the user for a list of numbers and prints out the maximum and minimum of the numbers at the end when the user enters "done". If the user enters anything other than a number, detect their mistake using try and except and print an error message and skip to the next number. The program should store the numbers the user enters in a list and use the max() and min() functions to compute the maximum and minimum numbers after the loop completes.

12) Develop an application that reads a file and prints the *letters* in decreasing order of frequency. Your program should convert all the input to lower case and only count the letters a-z. Your program should not count spaces, digits, punctuation, or anything other than the letters a-z. Also, print the top N (say N=10) words.

13) Develop an application to open the file (say, romeo.txt) and read it line by line. For each line, split the line into words using the *split* function. For each word, check to see if the word is already in a list. If the word is not in the list, add it to the list. When the program completes, sort and print the resulting words in alphabetical order.

14). Develop an application to read through the email data and when you find line that starts with "From", you will split the line into words using the split function. We are interested in who sent the message, which is the second word on the From line.
From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008
You will parse the From line and print out the second word for each From line, then you will also count the number of From (not From:) lines and print out a count at the end.

15) Develop an application to look for lines of the form, *(say, "New Revision: 39772")* and extract the number from each of the lines using a regular expression and the findall() method. Compute the average of the numbers and print out the average.

3. SPECIFIC LEARNING OUTCOMES (SLO)

<table>
<tr><th>EX. NO</th><th>CONTENT</th><th>Blooms Taxonomy Highest Level of Transaction</th></tr>
<tr><td>1.</td><td>Practice using the Python interpreter / Jupyter Notebook as a calculator:
A. The volume of a sphere with radius r is 4/3πr3.
B. What is the volume of a sphere with radius 5?
Suppose cover price of a book is INR 24.95, but a bookstore offers a 40% discount. Shipping costs INR 3.00 for the first copy and 75 Paise for each additional copy. What is the total wholesale cost for 60 copies?
If I leave my house at 6:52 am and run 1 km at an easy pace (8:15 per km), then 3 km at tempo (7:12 per km) and 1 km at easy pace again, what time do I get home for breakfast?</td><td>K5</td></tr>
<tr><td>2.</td><td>Develop an application to prompt for a score between 0.0 and 1.0. If the score is out of range, print an error message. If the score is between 0.0 and 1.0, print a grade using the following table:
<table>
<tr><td>Score</td><td>Grade</td><td>Score</td><td>Grade</td><td>Score</td><td>Grade</td><td>Score</td><td>Grade</td><td>Score</td><td>Grade</td></tr>
<tr><td>>= 0.9</td><td>A</td><td>>= 0.8</td><td>B</td><td>>= 0.7</td><td>C</td><td>>= 0.6</td><td>D</td><td>< 0.6</td><td>F</td></tr>
</table>
</td><td>K5</td></tr>
</table>

3.	Develop an application that repeatedly reads numbers until the user enters “done”. Once “done” is entered, print out the total, count, and average of the numbers. If the user enters anything other than a number, detect their mistake using try and except and print an error message and skip to the next number.	K5
4.	The **datetime** module provides time objects that are similar to the given objects, but they provide a rich set of methods and operators. Write a program that takes a birthday as input and prints the user’s tag and the number of days, hours, minutes and seconds until their birth.	K5
5.	Given a list of strings, return the count of the number of strings here the string length is 2 or more and the first and last chars of the string are the same.	K5
6.	Develop a Python function front_x(). Given a list of strings, return a list with the strings in sorted order, except group all the strings that begin with 'x' first. Eg. ['mix', 'xyz', 'apple', 'xanadu', 'aardvark'] yields Tamilnadu', 'xyz', 'aardvark', 'apple', 'mix']. Hint: this can be done by king 2 lists and sorting each of them before combining them.	K5
7.	Develop a Python function sort_last(). Given a list of non-empty tuples, return a list sorted in increasing order by the last element in each tuple. E.g. [(1, 7), (1, 3), (3, 4, 5), (2, 2)] yields [(2, 2), (1, 3), (3, 4, 5), (1, 7)]. Hint: use a custom key= function to extract the last element form each tuple.	K5
8.	Develop a Python function remove_adjacent(). Given a list of numbers, return a list where all adjacent same elements have been reduced to a single element, so [1, 2, 2, 3] returns [1, 2, 3]. List [2, 2, 3, 3] returns [2, 3]. List [ ] returns [ ]. You may create a new list or identiify the passed in list.	K5
9.	write a function verbing(). Given a string, if its length is at least 3, 'ing' to its end. Unless it already ends in 'ing', in which case add instead. If the string length is less than 3, leave it unchanged. turn the resulting string. So ‘hail’ yields: hailing; ‘swimming’ yields: immingly; ‘do’ yields: do.	K5
10.	Develop a function not_bad(). Given a string, find the first appearance of the substring 'not' and 'bad'. If the 'bad' follows the replace the whole 'not'...'bad' substring with 'good'. Return the resulting string. So 'This dinner is not that bad!' yields: This dinner is not bad!	K5
11.	Develop an application that prompts the user for a list of numbers and prints out the maximum and minimum of the numbers at the end when the user enters “done”. If the user enters anything other than a number, detect their mistake using try and except and print an error message and skip to the next number. The program should store the numbers the user enters in a list and use the max() and min() functions to compute the maximum and minimum numbers after the loop completes.	K5
12.	Develop an application that reads a file and prints the *letters* in decreasing order of frequency. Your program should convert all the input to lower case and only count the letters a-z. Your program should not count spaces, digits, punctuation, or anything other than the letters a-z. Also, print the top N (say N=10) words.	K5
13.	Develop an application to open the file (say, romeo.txt) and read it by line. For each line, split the line into words using the *split*	K6

	action. For each word, check to see if the word is already in a list. If the word isnot in the list, add it to the list. When the program upletes, sort and print the resulting words in alphabetical order.	
14.	Develop an application to read through the email data and when you find line that starts with “From”, you will split the line into words using the split function. We are interested in who sent the message, which is the second word on the From line. From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008 y You will parse the From line and print out the second word for each From line, then you will also count the number of From (not From:) lines and print out a count at the end.	K6
15.	Develop an application to look for lines of the form, *(say, “New Revision: 39772”)* and extract the number from each of the lines using a regular expression and the findall() method. Compute the average of the numbers and print out the average.	K6

4. MAPPING SCHEME(CO, PO, PSO)

L-Low **M-Moderate** **H- High**

U21CS2P2	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2	PSO3	PSO4
CO1	H	M	M	M	L	L	-	-	M	H	-	L	M
CO2	H	M	M	M	M	L	-	-	M	H	-	L	M
CO3	H	H	M	M	M	M	-	-	M	H	-	M	M
CO4	H	H	M	M	M	M	-	-	M	H	-	H	H
CO5	H	M	M	M	M	M	-	-	M	H	M	H	H
CO6	H	H	H	H	H	M	-	-	M	H	M	H	H

5. COURSE ASSESSMENT METHODS

DIRECT:

1. Continuous Assessment Test: T1, T2 (Practical Components): Closed Book
2. Cooperative Learning Report, Assignment, Group Discussion, project Report, Field Visit Report, Seminar.
3. Pre/Post Test, Viva, Report for each Exercise.
4. Lab Model Examination & End Semester Practical Examination

INDIRECT:

2. Course end survey (Feedback)

Name of the Course Coordinator: Prof. S. Maheswari

SBEC -1: BUSINESS COMMUNICATION AND COLLABORATION TOOLS

SEMESTER: II **CODE: U21CSPS1**
CREDITS : 2 **HOURS /WEEK: 2**

----

1. COURSE OUTCOMES

CO #	Course Outcomes	Level	Ex. No.
CO1	Apply the text manipulation techniques like font style, font size, bullets, headers and footers	K4	1-2
CO2	Create and edit a table and apply table styles and formatting	K5	3-4
CO3	Practice the concept of Mail Merge	K5	5
CO4	Solve real-time problems like payroll calculation and grade calculation	K5	6-7
CO5	Design the and electricity bill generation	K6	8
CO6	Analyze the data using different charts like bar chart, pie chart, etc.	K5	9-10

2. A. SYLLABUS

1. Prepare a Bio-Data to apply for job in IT Company using the text manipulation techniques like font style, font size, bullets, footers and headers.
2. Create a mark sheet for 10 students containing the following fields: name, roll no., mark1, mark2 and mark3 and total.
3. Design an effective Class Time Table.
4. Create mathematical formulae.
5. Write a letter to invite your friends for a get together using Mail Merge option.
6. Tabulate a list of 10 students and grade them based on their seminar performance and then sort them according to their grades
7. Prepare a payroll for 10 employees of a company using the following details.
   - If the basic salary is between 15000 and 20000, then 20% of basic salary is given to HRA, 25% to DA, 12% to LIC and 15% to PF.
   - If the basic salary is less than 15000, then 25% is given to HRA, 30% to DA, 11% to LIC and 16% to PF.
   - If the basic salary is more than 20000, then 15 % is given to HRA, 40% to DA, 14% to LIC and 15% to PF Compute the GrossPay, Deductions and NetPay.
8. Prepare a Electricity Bill for the following table:

	1-500 Units	501-800 Units	>800 Units
House	6	8	11.50
Business Enterprise	9	12	17.50
Education Institution	7	10	13.25

   Compute the amount to be paid to the Electricity Board.

9. Prepare a Bar chart for analyzing the admission of students in your college for the past five years.
10. Prepare a pie chart for scheduling your daily activities.

3. SPECIFIC LEARNING OUTCOMES

<table>
<tr><th>EXP NO</th><th>LEARNING OUTCOMES</th><th>Blooms Taxonomy Highest Level of Transaction</th></tr>
<tr><td>1.</td><td>Prepare a Bio-Data to apply for job in IT Company using the text manipulation techniques like font style, font size, bullets, footers and headers.</td><td>K3</td></tr>
<tr><td>2.</td><td>Create a mark sheet for 10 students containing the following fields: name, roll no., mark1, mark2 and mark3 and total.</td><td>K3</td></tr>
<tr><td>3.</td><td>Design an effective Class Time Table.</td><td>K6</td></tr>
<tr><td>4.</td><td>Create mathematical formulae.</td><td>K3</td></tr>
<tr><td>5.</td><td>Write a letter to invite your friends for a get together using Mail Merge option.</td><td>K6</td></tr>
<tr><td>6.</td><td>Tabulate a list of 10 students and grade them based on their seminar performance and then sort them according to their grades.</td><td>K5</td></tr>
<tr><td>7.</td><td>Prepare a payroll for 10 employees of a company using the following details.
<ul>
<li>If the basic salary is between 15000 and 20000, then 20% of basic salary is given to HRA, 25% to DA, 12% to LIC and 15% to PF.</li>
<li>If the basic salary is less than 15000, then 25% is given to HRA, 30% to DA, 11% to LIC and 16% to PF.</li>
<li>If the basic salary is more than 20000, then 15 % is given to HRA, 40% to DA, 14% to LIC and 15% to PF Compute the GrossPay, Deductions and NetPay.</li>
</ul></td><td>K3</td></tr>
<tr><td>8.</td><td>Prepare a Electricity Bill for the following table:
<table>
<tr><th></th><th>1-500 Units</th><th>501-800 Units</th><th></th></tr>
<tr><td>House</td><td>6</td><td>8</td><td></td></tr>
<tr><td>Business Enterprise</td><td>9</td><td>12</td><td></td></tr>
<tr><td>Education Institution</td><td>7</td><td>10</td><td></td></tr>
</table></td><td>K3</td></tr>
<tr><td>9.</td><td>Prepare a Bar chart for analyzing the admission of students in your college for the past five years</td><td>K4</td></tr>
<tr><td>10.</td><td>Prepare a pie chart for scheduling your daily activities.</td><td>K4</td></tr>
</table>

4. MAPPING SCHEME (CO, PO & PSO)

U21CSSP1	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2	PSO3	PSO4
CO1	H	M	M	M	L	L	-	-	M	H	-	L	M
CO2	H	M	M	M	M	L	-	-	M	H	-	L	M
CO3	H	H	M	M	M	M	-	-	M	H	-	M	M
CO4	H	H	M	M	M	M	-	-	M	H	-	H	H
CO5	H	M	M	M	M	M	-	-	M	H	M	H	H
CO6	H	H	H	H	H	M	-	-	M	H	M	H	H

5. COURSE ASSESSMENT METHOD

DIRECT:

1. Continuous Assessment Test: T1, T2 (Practical Components): Closed Book
2. Cooperative Learning Report, Assignment, Group Discussion, project Report, Field Visit Report, Seminar.
3. Pre/Post Test, Viva, Report for each Exercise.
4. Lab Model Examination & End Semester Practical Examination

INDIRECT:

1. Course end survey (Feedback)

Name of the Course Coordinator: Dr. M. Jayakkumar

CORE III: OBJECT ORIENTED PROGRAMMING CONCEPTS

SEMESTER: III **CODE: U21CS303**

CREDITS : 5 **HOURS /WEEK: 6**

----

1. COURSE OUTCOMES

CO #	Course Outcomes	Level	Unit
CO1	Apply the concepts of the variables, constants, arrays and operators for various problems	K5	1
CO2	Apply various input and output functions for getting values and displaying the result	K5	2
CO3	Explain the usage of control structures like decision statements and looping statements	K5	3
CO4	Analyze the user-defined and built-in functions for various modules	K6	4
CO5	Analyze the object oriented concepts like classes, objects, constructors and create the real time applications using inheritance	K6	5
CO6	Create GUI based applications through applets	K6	5

3. A. SYLLABUS

Unit I : Java Fundamentals and decisions 15

Parts of Java program, println and print methods, Escape sequences - Variables, literals and primitive data types - Arithmetic operators, expressions, precedence and evaluation of expressions - Type conversion, Math class, String class and comment statements - Reading keyboard input: Scanner class and its methods - Reading keyboard input: Message dialog box - Reading keyboard input: Input dialog box - Converting String inputs to numbers - Relational operators, if, if-else, if-else-if statements - Logical operators and its precedence, String comparison and Conditional operator.

Unit II : Multi point selection and Loops 15

Switch statement, String cases for switch statement - Arithmetic calculator application - Format specifications using printf method, Increment and decrement operators - While loop, indefinite loop, while loop for input validation - Do-while loop - For loop, variable declaration in for loop, multiple statements in initialization - Running totals and sentinels - Nested loops - Break and continue statement, Deciding which loop to use - Exception handling using try-catch-finally statements.

Unit III : Methods and Arrays 15

Method Definition: Void and value-returning methods, Calling a method, Method calling another method - Passing single argument to a method - Passing multiple arguments to a method, Passing object references to a method, Local variables - Defining and calling value returning method - Returning Boolean value, Returning a reference to an object - Problem solving with methods : Divide and conquer technique - Arrays: declaration, accessing elements, inputting and outputting array contents, Bounds checking, Array initialization - Processing array elements: Array length, Enhanced For loop, Copying arrays - Passing arrays as arguments, Returning arrays from methods, Array of objects - 2D arrays: Initialization, Displaying elements, Summing rows and columns, Passing 2D arrays to methods, Multi-dimensional arrays.

Unit IV : Object Oriented Programming using Java 15

Objects, classes, Writing a simple class and its methods, Showing access specification in UML diagram - Instance variables and methods, Constructors, default constructor - Overloading methods and constructors - Static variables, Static methods - Passing objects as arguments to methods, Returning objects from methods – Aggregation - Inheritance: Generalization & specialization, IS-A relationship - Calling superclass constructor, Overriding superclass methods, Protected members - Abstract classes and abstract methods – Interfaces.

Unit V : Data structures, Files and Applets 15

Stack - String Tokenizer, Hashtable - Writing data to file using PrintWriter class - Reading text and primitive data from file using Scanner's nextLine method - FileOuputStream and FileInputStream classes - FileWriter and FileReader classes - Creating and running Applets - Drawing Lines, Rectangles and Ovals in Applets - Drawing Arcs and Polygons in Applets - Playing Audio clips in Applets.

B. TOPICS FOR SELF STUDY

S.No.	Topics	Web Links
1	Uml Class diagram	https://www.youtube.com/watch?v=UI6lqHOVHic
2	Entity Diagram	https://www.youtube.com/watch?v=QpdhBUYk7Kk
3	Sql Applications	https://www.youtube.com/watch?v=dPx1228_EYM
4	String Tokenizer	https://www.youtube.com/watch?v=goXyTvV1M40

C. TEXT BOOK

1. Tony Gaddis. "***Starting out with Java: From control structures through objects"***, 5ed, Pearson, 2013

D. REFERENCE BOOKS

1. Stuart Reges and Marty Stepp. "”’***Building Java programs***: Back to basics approach, 3ed, Pearson, 2014.
2. Robert Sedgewick and Kevin Wayne. "***Introduction to Programming in Java: An Interdisciplinary Approach***", 1e, Pearson, 2014

E. WEB LINKS

- https://www.coursera.org/courses?query=advanced%20java
- https://www.learnvern.com/course/advanced-java-tutorial
- https://nareshit.in/advanced-java-training/

3. SPECIFIC LEARNING OUTCOMES (SLO)

Unit	Course Content	Learning outcomes	Blooms Taxonomy Level
I	Java Fundamentals and decisions		
1.1	Data types and Variables	1. Recall the different data types.	[K1]

		2. Create variable for real life entities(eg: age, height, weight)	
1.2	I/O Statements	1. Classify different types of Input functions to get a value of different data types of variables.(such as nextInt, nextFloat etc.,) 2. Apply print statements for both numeric and alphanumeric characters.	[K2,K3]
1.3	Operators	1. Categorize the different types of operators.	[K4]
1.4	Decision Statements	1. Construct alternative statements to find the biggest value of n values.	[K6]
1.5	Math Class	1. Apply various mathematical functions(eg: max, min, abs etc.,)	[K3]
1.6	String Class	1. List out the string functions 2. Apply various string functions(eg: length, toUpperCase, charAt etc.,)	[K3,K4]
II	**Multi point selection and Loops**		
2.1	Switch Statement	1. Constructs branches of conditions of a program. 2. Compare Switch statement and if-else statement.	[K3,K4]
2.2	Format specifications	Apply the format specification statement to get an output in various ways.(eg: System.out.printf) statement.	[K3]
2.3	Looping Statements	1.Compare different looping statements (such as For, while and Do.. while) 2.Categorize Different types of Looping 2. Apply looping statements for a continuous process.	[K2-K5]
2.4	Exception Handling	Apply exception handling statements to the debug process.	[K3]
III	**Methods and Arrays**		
3.1	Method	1. Recollect the syntax for define a function 2. Classify different types of functions 3. Examine calling and returning a function 4. Develop a function to swap two values	[K1-K6]
3.2	Variables	1. Contrast local and global variables	[K3]
3.3	Arrays	1. Explain Array variables 2. Illustrate memory allocation for arrays and accessing an array elements.	[K2,K3]
3.4	Problem Solving Methods	Apply divide and conquer techniques for solving real time complex problems.	[K3]
IV	**Object Oriented Programming using Java**		
4.1	Class and Object	1. Create a class and objects for an entity(eg: Human, Car etc.,) 2. Analyze the behavior of an object,Constructors	[K4-K6]
4.2	Static Variable and Methods	Create static variables and methods of a class.	

			[K6]
V	**Data structures, Files and Applets**		
5.1	String Tokenanizer	1.Summarize String Tokenanizer	[K2]
5.2	Stack	1.Apply the stack concepts and evaluate if the given expression is balanced or not.	[K3,K5]
5.3	Files	1. Apply various types of read and write streams to be used to get a value from the local directory. 2. Analyze the contents placed in the file.	[K3,K4]
5.4	Applets	Create a GUI using Applet controls.	[K6]

4. MAPPING SCHEME(CO, PO, PSO)

L-Low M-Moderate H- High

U21CS303	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2	PSO3	PSO4
CO1	H	M	M	M	L	L	-	-	M	H	-	L	M
CO2	H	M	M	M	M	L	-	-	M	H	-	L	M
CO3	H	H	M	M	M	M	-	-	M	H	-	M	M
CO4	H	H	M	M	M	M	-	-	M	H	-	H	H
CO5	H	M	M	M	H	M	-	-	M	H	M	H	H
CO6	H	H	H	H	H	M	-	-	M	H	M	H	H

5. COURSE ASSESSMENT METHODS

DIRECT:

1. Continuous Assessment Test: T1, T2 (Theory & Practical Components): Closed Book
2. Open Book Test.
3. Cooperative Learning Report, Assignment, Group Presentation, Group Discussion, project report, Field Visit Report, Poster Presentation, Seminar, Quiz (written).
4. Pre-Semester & End Semester Theory Examination

INDIRECT:

1.Course end survey (Feedback)

Name of the Course Coordinator: Prof. C. Linda Hepsiba

CORE PRACTICAL III: OBJECT ORIENTED PROGRAMMING CONCEPTS LAB

SEMESTER: III **CODE: U21CS3P3**

CREDITS: 2 **HOURS /WEEK: 3**

---

1. COURSE OUTCOMES

CO #	Course Outcomes	Level	Exercise No.
CO1	Develop the applications for simple use-cases using preliminary concepts	K5	I: 1-7
CO2	Design the scientific calculator using the java operators	K6	II: 1
CO3	Develop the applications for employee roll and students' mark list	K6	II: 2:3
CO4	Develop the applications for matching the brackets	K6	II: 4
CO5	Design the applications for counting the words in text files	K6	II: 5
CO6	Develop an application with GUI by having Applets and Swing.	K6	II: 6

2. A. SYLLABUS

Part-I: Simple Applications **15**

1. Read salaries (*double* values) of three employees and display the highest salary
2. Calculate the grade for a given mark of a student. Grade is A if mark >= 80, B if mark in between 60 and 79, C if mark in between 40 and 59, F if mark less than 40)
3. Check an year leap or not
4. Print the multiplication table of order **N×N**
5. Generate the first ***n*** Prime numbers
6. Calculate Factorial of a given number
7. Create an array ***temp*** and read into this array 7 temperature values (in Celsius) of a city, representing the weather from Monday to Sunday and display which day was the hottest in that week.

Part-II: Advanced Applications **15**

1)Scientific Calculator: Develop an application that performs the following operations.

- Addition, subtraction, multiplication, division and reminder
- Sin(x), Cos(x), Tan(x)
- Log(x), e^x, 2^x

Note: The application should repeatedly perform operations until a user quits the program, based on some condition (eg. operator == '#')

2)Top Employee: Develop an application that reads salaries of N employees in **M** departments and perform the following operations

- Display the average salary of each employee
- Display the top employee, the one whose average salary is the highest

3) Students Mark List: Create a class *Student* with *roll no, name* and 3 marks (say, *m1*, *m2*, *m3*). Create a *constructor* that assigns the values of roll no, name and 3 marks. Then, create a method ***display()*** that displays the rollno, name and 3 marks. Create another method ***calculate_ result()*** that prints the result as either *"Pass"* or *"Fail"*. The student has passed all exams if he secures at least 40 marks in each subject. Create a public class *Mark List* that instantiates ***N*** student objects by reading roll no, name and marks and displays the details and result of those students.

4) Brackets Matcher: Develop an application that reads a sequence of characters, and determines whether its parentheses, braces, and curly braces are balanced. (Hint: for left delimiters, push onto Stack; for right delimiters, pop from Stack and check whether popped element matches right delimiter).

5)Word Counter: Develop an application to read the contents of a large text file and count the occurrences of each word (Hint: use String Tokenizer to parse the file and Hashtable/ HashMap to store the words and their count, also use File Reader for reading the file contents).

6)Smiley face Applet: Create an applet that will display 3 smiley faces in three different colors.

3. SPECIFIC LEARNING OUTCOMES (SLO)

Unit/ Section	Course Content	Blooms Taxonomy Level
I	**Part -1 Simple Applications**	
1	Read salaries (double values) of three employees and display the highest salary	K6
2	Calculate the grade for a given mark of a student. Grade is A if mark >= 80, B if mark in between 60 and 79, C if mark in between 40 and 59, F if mark less than 40)	K6
3	Check an year leap or not	K6
4	Print the multiplication table of order **N x N**	K6
5	Generate the first **n** Prime numbers	K6
6	Calculate Factorial of a given number	K6
7	Create an array **temp** and read into this array 7 temperature values (in Celsius) of a city, representing the weather from Monday to Sunday and display which day was the hottest in that week	K6
	Part-II: Advanced Applications	
1	**Scientific Calculator:** Develop an application that performs the following operations. ● Addition, subtraction, multiplication, division and reminder ● Sin(x), Cos(x), Tan(x) ● Log(x), e^x, 2^x **Note:** The application should repeatedly perform operations until a user quits the program, based on some condition (eg. operator == '#')	K6
2	**Top Employee:**	K6

	Develop an application that reads salaries of N employees in **M** departments and perform the following operations ● Display the average salary of each employee ● Display the top employee, the one whose average salary is the highest	
3	**Students Mark List:** Create a class Student with roll no, name and 3 marks (say, m1, m2, m3). Create a constructor that assigns the values of roll no, name and 3 marks. Then, create a method **display ( )** that displays the roll no, name and 3 marks. Create another method **calculate_ result ( )** that prints the result as either “Pass” or “Fail”. The student has passed all exams if he secures at least 40 marks in each subject. Create a public class Mark List that instantiates **N** student objects by reading roll no, name and marks and displays the details and result of those students.	K6
4	**Brackets Matcher:** Develop an application that reads a sequence of characters, and determines whether its parentheses, braces, and curly braces are balanced. (Hint: for left delimiters, push onto Stack; for right delimiters, pop from Stack and check whether popped element matches right delimiter).	K6
5	**Word Counter:** Develop an application to read the contents of a large text file and count the occurrences of each word (Hint: use String Tokenizer to parse the file and Hash table/Hash Map to store the words and their count, also use File Reader for reading the file contents).	K6
6	**Smiley face Applet**: Create an applet that will display 3 smiley faces in three different colors.	K6

4. MAPPING (CO, PO, PSO)

L-Low **M-Moderate** **H- High**

U21CS3P3	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2	PSO3	PSO4
CO1	H	M	M	M	L	L	-	-	M	H	-	L	M
CO2	H	M	M	M	M	L	-	-	M	H	-	L	M
CO3	H	H	M	M	M	M	-	-	M	H	-	M	M
CO4	H	H	M	M	M	M	-	-	M	H	-	H	H
CO5	H	M	M	M	H	M	-	-	M	H	M	H	H
CO6	H	H	H	H	H	M	-	-	M	H	M	H	H

5. COURSE ASSESSMENT METHODS

DIRECT:

1. Continuous Assessment Test: T1, T2 (Theory & Practical Components): Closed Book
2. Open Book Test.
3. Cooperative Learning Report, Assignment, Group Presentation, Group Discussion, project report, Field Visit Report, Poster Presentation, Seminar, Quiz (written).
4. Pre-Semester & End Semester Theory Examination

INDIRECT:

1.Course end survey (Feedback)

Name of the Course Coordinator: Dr. C. Linda Hepsiba

Core IV: DESIGN & IMPLEMENTATION OF DATABASE

SEMESTER: IV **CODE:U21CS404**
CREDIT:5 **HOURS /WEEK: 6**

---

1. COURSE OUTCOMES

CO #	Course Outcomes	Level	Unit
CO1	Design the SQL Queries for selecting and sorting the data from table	K6	1
CO2	Design the SQL sub queries using Join, view and index for table	K6	2
CO3	Apply the DB system development life cycle to business problems	K5	3
CO4	Build the table to create, modify data into table and remove data from the table.	K6	4
CO5	Access the ER diagram for representing conceptual data model.	K5	4
CO6	Adapt the ideas of normalization to handle different kinds of forms	K5	5

2. A. SYLLABUS

Unit I: Introduction and Structured Query Language Part-I **15**

Physical file based system versus database system approaches – DBMS: terminologies, components, roles, advantages and disadvantages – Database architectures: teleprocessing, file server, 2-tier, 3-tier, N-tier, middleware and Transaction processing monitor – Software components of DBMS and Database Manager – Objectives of SQL and BNF notation to define SQL statements – SELECT statement to retrieve all rows – Selecting rows using WHERE clause – Sorting result using ORDERBY clause – SQL aggregate functions – Grouping results using GROUPBY clause and HAVING clause.

Unit II: Structured Query Language Part-II **15**

Designing Sub Queries – Using keywords ANY and ALL – JOIN – inner and outer joins for joining relations – EXISTS and NON EXISTS keywords – Combining results of queries using UNION, INTERSECT and EXCEPT – Updating databases using INSERT, UPDATE and DELETE – SQL data types and Integrity control – Creating, altering and removing tables – Indexes: CREATE and REMOVE. Views: CREATE and REMOVE

Unit III: DBS development life cycle and ER modeling **15**

DBS development lifecycle stages and their activities – DBS development: Requirements analysis – DBS development: Database design – DBS development: DBMS selection and Application design – ER diagram: Entity types and relationship types – Attributes types and keys of ER diagram – Strong and weak entity types; attributes on relationships – Relationship types: One-to-one, One-to-many, Many-to-many – N-ary relationships, cardinality and participation concepts – Connection traps.

Unit IV: Normalization and Conceptual DB design **15**

Purpose of normalization and its support for DB design – Update anomalies: problems related to redundant data – Functional dependencies: characteristics and identification – Normalizations: 1NF, 2NF and 3NF –

Normalization Example – Overview of DB Design Methodology – Conceptual DB Design Methodology: building conceptual data model (ie. ER diagram) – Identifying entity types, relationship types and attribute types – Determining domain and key attributes (candidate, primary and alternate) – Checking for redundancy, validating ER diagram and reviewing ER diagram with users.

Unit V: Logical Database design **15**

Logical DB Design Methodology: building logical data model (ie. Relational schema) – Creating relations: for strong entity types and weak entity types – Creating relations: for 1-M binary, 1-1 Binary and 1-1 Recursive relationships – Creating relations: Super class/sub class relationship types – Creating relations: for M-M Binary relationships – Creating relations: for complex relationship types and multi valued attributes – Validating relations using normalization – Checking integrity constraints – Reviewing logical data model with user – Merging logical data models into global model.

B. TOPICS FOR SELF STUDY

S. No.	Topics	Web Links
1	DBMS QB	https://www.youtube.com/watch?v=ztHopE5Wnpc
2	System Design	https://www.youtube.com/watch?v=KmAyPUv9gOY
3	Message Queue	https://www.youtube.com/watch?v=oUJbuFMyBDk
4	Binary Relationship	https://www.youtube.com/watch?v=xmDYjXAEi1w

C. TEXT BOOK

1. Thomas M. Connolly and Carolyn E. Begg. Database Systems: “***A Practical Approach to Design, Implementation, and Management”,*** 6th Edition, Pearson, 2015. **(Chapters 1, 3.1, 3.6, 6, 7, 10, 12, 14, 16, 17 only)**

D. REFERENCES BOOKS

1. Hoffer, J. A., Venkataraman, R., and Topi, H. “***Modern database management”*** (12th ed.), Pearson, 2016.
2. Coronel, Morris and Rob. “***Database Systems: Design, Implementation and Management”***, 12thed, Cengage Learning, 2017.

E. WEB LINKS

- https://www.edx.org/course/advanced-database-administration
- https://www.udemy.com/course/introduction-to-databases-and-sql-querying/
- https://www.mygreatlearning.com/academy/learn-for-free/courses/database-management-systems-dbms

3. SPECIFIC LEARNING OUTCOMES (SLO)

Unit	Topics	Unit Learning Outcomes	Level
I	**Introduction and Structured Query Language Part-I**		
1.1	Physical file based system versus database system approaches	Compare between file based system and database system	K2

1.2	DBMS: terminologies, components, roles, advantages and disadvantages	1.Identify the terminologies of DBMS 2.List the roles of database 3.Summarize the advantages of DBMS	[K1-K3]
1.3	Database architectures: teleprocessing, file server, 2-tier, 3-tier, N-tier, middleware and Transaction processing monitor -- Software components of DBMS and Database Manager	-Build the architecture of database -Classify the software components	[K2,K3]
1.4	Objectives of SQL and BNF notation to define SQL statement	Define the SQL statements -Make use of BNF notation	K3
1.5	SELECT statement to retrieve all rows	-How to retrieve all rows using select statements -Apply where clause for selecting rows	K5
1.6	Sorting result using ORDERBY clause – Grouping results using GROUPBY clause and HAVING clause	-Create orderby clause for sort the values -Create groupby clause for group the values	K3
1.7	SQL aggregate functions	Analyze the various aggregate functions	K4
II	**Structured Query Language Part-II**		
2.1	Designing Sub Queries – Using keywords ANY and ALL	-Build a sub queries Compare the queries any and all	K4
2.2	JOIN – inner and outer joins for joining relations – EXISTS and NON EXISTS keywords	Apply the join queries	K3
2.3	Combining results of queries using UNION, INTERSECT and EXCEPT – Updating databases using INSERT, UPDATE and DELETE	-Combine the queries -Apply the union, intersect and except in queries -Modify the database	K4
2.4	SQL data types and Integrity control – Creating, altering and removing tables	-Recall data type and integrity control - Changing and deleting the tables	K4
2.5	Indexes: CREATE and REMOVE. Views: CREATE and REMOVE	-Apply index and view -Create and remove the index and view	K5
III	**DBS development life cycle and ER modeling**		
3.1	DBS development lifecycle stages and their activities – DBS development: Requirements analysis	-Recall the life cycle stages - Apply life cycle activities -Analyze the DBS development	K4
3.2	DBS development: DBMS selection and Application design	-Define DBS development - Choose the DBMS design - Apply the database design	K3
3.3	ER diagram: Entity types and relationship types – Attributes types and keys of ER diagram	-List the entity types - Develop the ER diagram - Explain the ER diagram	K5
3.4	Strong and weak entity types	Identify the strong and weak entity types	K3
3.5	Relationship types: One-to-one, One-to-many, Many-to-many – N-ary relationships, cardinality and participation concepts	-Explain the relationship types - Organize the relationship Analyze the relationship with participation concept	K4
3.6	Connection traps	-Create a connection traps	K4
IV	**Normalization and Conceptual DB design**		

4.1	Purpose of normalization and its support for DB design	1.Explain normalization 2. Apply normalization in DB	K1,K3
4.2	Update anomalies: problems related to redundant data	Modify anomalies	K5
4.3	Functional dependencies: characteristics and identification	-Identify the functional dependencies	K4
4.4	Normalizations: 1NF, 2NF and 3NF – Normalization Example	Classify normalization -Relate normal forms -Select the correct normal forms	K4
4.5	Overview of DB Design Methodology – Conceptual DB Design Methodology: building conceptual data model	Build the conceptual data model Explain the DB methodology	K4
4.6	Identifying entity types, relationship types and attribute types Determining domain and key attributes	1.Identify the ER Type 2.Relate entity types and relationship type 3.Categorize the attribute types 4.Determine the domain and key attributes	K1-K5
4.7	– Checking for redundancy, validating ER diagram and reviewing ER diagram with users.	-Test the redundancy -Evaluate ER diagram	K5
V	**Logical Database design**		
5.1	Logical DB Design Methodology: building logical data model	Build the logical model	K4
5.2	– Creating relations: for strong entity types and weak entity types	Create relations for entity types	K5
5.3	Creating relations: for 1-M binary, 1-1 Binary and 1-1 Recursive relationships	Create relations for binary and recursive relationship	K5
5.4	Creating relations: Super class/sub class relationship types	Create relations for super and sub class	K5
5.5	Creating relations: for M-M Binary relationships – Creating relations: for complex relationship types and multi valued attributes	Create relations for multi valued attributes	K5
5.6	Validating relations using normalization – Checking integrity constraints	-Test the integrity constraint -validate the relations	K5
5.7	Reviewing logical data model with user – Merging logical data models into global model	Combine the logical model into global model	K5

4. MAPPING SCHEME (CO, PO, PSO)

L-Low M-Moderate H- High

U21CS404	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2	PSO3	PSO4
CO1	H	M	M	M	L	L	-	-	M	H	-	L	M
CO2	H	M	M	M	M	L	-	-	M	H	-	L	M
CO3	H	H	M	M	M	M	-	-	M	H	-	M	M
CO4	H	H	M	M	M	M	-	-	M	H	-	H	H
CO5	H	M	M	M	H	M	-	-	M	H	M	H	H
CO6	H	H	H	H	H	M	-	-	M	H	M	H	H

5. COURSE ASSESSMENT METHODS

DIRECT:

1. Continuous Assessment Test: T1, T2 (Theory & Practical Components): Closed Book
2. Open Book Test.
3. Cooperative Learning Report, Assignment, Group Presentation, Group Discussion, project report, Field Visit Report, Poster Presentation, Seminar, Quiz (written).
4. Pre-Semester & End Semester Theory Examination

INDIRECT:

1.Course end survey (Feedback)

Name of the Course Coordinator : R. Cynthia Monica Priya

CORE PRACTICAL IV: DESIGN & IMPLEMENTATION OF DATABASE LAB

SEMESTER: 4 **COURSE CODE: U21CS4P4**
CREDITS: 2 **HOURS /WEEK: 3**

---

1. COURSE OUTCOMES

CO #	COURSE OUTCOMES	LEVEL	EX. NO.
CO1	Define the basic DDL commands(CREATE ALTER, DROP, TRUNCATE, COMMIT, ROLLBACK and RENAME) in RDBMS	K5	1
CO2	Build the commands of DML and DCL(INSERT, UPDATE, DELETE, MERGE, LOCK, grant and revoke) the table value in RDBMS	K5	2-3
CO3	Create the Cursers, Procedures and Triggers for Database	K6	4-6
CO4	Demonstrate the different kind of embedded systems in SQL.	K5	7
CO5	Create Student information system such as rollno, name, class, mark, total, average and grade and implement in SQL form builder	K5	8
CO6	Develop Employee pay roll processing system and Banking System in sql form builder	K5	9-10

2. SYLLABUS

1. DDL commands in RDBMS.
2. DML and DCL commands in RDBMS Part-I.
3. DML and DCL commands in RDBMS Part-II.
4. High level language extensions with cursers.
5. High level language extensions with triggers.
6. Procedures and triggers.
7. Embedded SQL.
8. Design and implementation of Student information system.
9. Design and implementation of Employee pay roll processing system.
10. Design and implementation of Banking system.

3. SPECIFIC LEARNING OUTCOMES (SLO)

Ex. No	Title of the Exercises	Level
1.	DDL commands in RDBMS	K1
2.	DML and DCL commands in RDBMS Part-I.	K3
3.	DML and DCL commands in RDBMS Part-II.	K3
4.	High level language extensions with cursers	K3

5.	High level language extensions with triggers	K3
6.	Procedures and triggers	K3
7.	Embedded SQL	K2
8.	Design and implementation of Student information system	K6
9.	Design and implementation of Employee pay roll processing system.	K6
10.	Design and implementation of Banking system	K6

4. MAPPING SCHEME (CO, PO, PSO)

L-Low M-Moderate H- High

U21CS4P4	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2	PSO3	PSO4
CO1	H	M	M	M	L	L	-	-	M	H	-	L	M
CO2	H	M	M	M	M	L	-	-	M	H	-	L	M
CO3	H	H	M	M	M	M	-	-	M	H	-	M	M
CO4	H	H	M	M	M	M	-	-	M	H	-	H	H
CO5	H	M	M	M	H	M	-	-	M	H	M	H	H
CO6	H	H	H	H	H	M	-	-	M	H	M	H	H

5. COURSE ASSESSMENT METHODS

DIRECT:

1. Continuous Assessment Test: T1, T2 (Practical Components): Closed Book
2. Assignment, Group Discussion, Project Report
3. Pre/Post Test, Viva, Report for each Exercise.
4. Lab Model Examination & End Semester Practical Examination

INDIRECT:

1.Course end survey (Feedback)

Name of Course Coordinator: Dr. R. Cynthia Monica Priya

CORE V: WEB PROGRAMMING

SEMESTER: V **CODE: U21CS505**

CREDITS: 5 **HOURS /WEEK: 5**

----

1. COURSE OUTCOMES

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

CO #	Course Outcomes	Level	Unit
CO1	Illustrate the components of HTML, CSS, Javascript and PHP in Web design.	K4	1
CO2	Categorize appropriate PHP functions and OOP concepts while coding.	K5	2
CO3	Distinguish MYSQL queries for the given problem statement and establish DB connection	K4	3
CO4	Access MYSQL using PHP.	K6	3
CO5	Validate the user inputs, identify user and track session using cookies and session.	K5	4
CO6	Design a website deploying the features of PHP and MYSQL.	K5	5

2. A. SYLLABUS

Unit I: Fundamentals of PHP

The structure of PHP, Basic syntax of PHP, Incorporating PHP within HTML - Using Comments, Variables, Variable Assignment - Operators, Multiple line commands - Constants, Predefined constants, The echo and print statements, variable scope – Functions - Expressions, Literals and Variables - Operator Precedence with Examples - Associativity, Relational Operators - Conditionals – ((if,else ..)– check minimum bank balance example)(switch – Menu selection example) - Looping (example for all loops – multiplication tables).

Unit II: PHP Fuctions and OOP

Define Function, Return types(Values, array) - Return global variables, variable scope - Include Statement, Using - (include once, require – once) - OOP – Declaring class, objects - Accessing objects, cloning objects, constructors, Destructor - Methods, Declaring constants - Inheritance - Arrays, For each loop, Accessing array elements - Using Array function - File Handling.

Unit III: MYSQL Basics

Database terms, Accessing MYSQL via command line - MYSQL commands – Datatypes – Indexes - Querying MYSQL database - Joining Tables - Using logical operators - MYSQL Functions - Simple example: Student details - Accessing MYSQL via PHPMYADMIN.

Unit IV: Accessing MYSQL using PHP

Querying MYSQL database through PHP - The $-POST array - Table Processing (Create, describe, drop,..) - Data Manipulation through PHP (Insert, update, select, delete), Using AUTO¬_INCREMENT - Simple Example: - Employee details - Performing additional queries - Using Mysqli procedurally - Form Handling: Building forms, Retrieving submitted data - Register globals, Default values, input types, sanitizing input - Example program: program to convert values between Farenheit to Celsius.

Unit V: Cookies, Sessions, Authentication and Validation

Using cookies in PHP - HTTP Authentication - Using Sessions - Validating user Input - Regular Expression, Matching through metacharacters, fuzzy character matching, grouping - Character classes, Indicating range, Negation - General Modifiers - Simple example: Tracking user session in online store, login validation - Using regular expression in PHP - Redisplaying a form after PHP validation.

B. TOPICS FOR SELF-STUDY

S.No.	Topics	Web Links
1	Angular JS	https://www.coursera.org/courses?query=angularjs
2	Node JS	https://www.coursera.org/courses?query=node%20js
3	React JS	https://reactjs.org/community/courses.html

C. TEXT BOOKS

1. Robin Nixon, "Learning PHP, MySQL & JavaScript With jQuery, CSS & HTML5", 5th Edition, O'Reilly Media, Inc., 2018.

D. REFERENCES

1. Steven Holzner, "PHP: The Complete Reference", McGraw-Hill, 1st Edition, 2007.
2. Luke Welling, Laura Thomson, "PHP and MySQL Web Development" , 4th Edition, Pearson Education, Inc, 2009.

E. WEB LINKS

- https://www.w3schools.com/php/
- https://www.tutorialspoint.com/php/index.htm
- https://www.javatpoint.com/php-tutorial

3. SPECIFIC LEARNING OUTCOMES

Unit	Topics	Unit Learning Outcomes	Level
	By the end of each topic of a unit, students will be able to		
I	**Fundamentals of PHP**		
	The structure of PHP, Basic syntax of PHP	Explain the basic structure and Define the syntax	K2
	Incorporating PHP within HTML	Demonstrate the integration process of PHP within HTML	K3
	Using Comments, Variables, Variable Assignment, Expressions, Literals and Variables	Practice different types of comments, identify Expressions, Literals, Variables and assign values to variables.	K5
	Operators, Multiple line commands	Use suitable operators for simple programs with multiple line commands	K3
	Constants, Predefined constants	Distinguish user defined and Predefined constants	K2
	Echo, print statements	Differentiate echo and print statements	K2
	Variable scope	Identify the scope of a variable	K2
	Functions	Define *functions* to separate out sections of code that perform a particular task.	K2

	Operator Precedence with Examples, Associativity, Relational Operators	Use operators based on their precedence and their associativity.	K3
	Conditionals – ((if,else ..), Looping	Apply conditionals and loops in programs.	K3
II	**PHP Functions and OOP**		
	Define Function	Create user defined functions and select predefined functions for the given problems.	K5
	Return types(Values, array)	Categorize the return types in functions	K4
	Return global variables	Discriminate global variables	K2
	Variable scope	Declare variables based on their scope	K3
	Include Statement, Using - (include once, require – once)	Differentiate include once and require once statements	K2
	OOP concepts	Apply OOP concepts in given problems.	K5
	Declaring class, objects	Declare class and define objects in programs.	K3
	Accessing objects, cloning objects, constructors, Destructor	Access the defined class using objects, define user defined constructor, and destructor.	K4
	Methods, Declaring constants	Write methods for the given problem with constants	K5
	Inheritance	Create programs with different types of inheritance for the given problems.	K4
	Arrays, For each loop	Include arrays for the given problem and define foreach loop	K3
	Accessing array elements	Access array elements using foreach loop	K5
	Using Array function	Manipulate arrays using different array functions	K5
	File Handling	Use file handling functions to create file, read file line by line, read file character by character, write file, append file, delete file and close file	K5
III	**MYSQL Basics**		
	Database terms	Explain and identify database terms	K2
	Accessing MYSQL via command line	Access MYSQL database through command line	K5
	MYSQL commands	Issue a wide range of commands from the mysql command-prompt to the database server such as creating and deleting databases and tables, searching for data, adding new rows and much more.	K5
	Datatypes, Indexes	Use MySQL data types effectively in designing databases in MySQL.	K5

		Create and use Indexes to find rows with specific column values quickly.	
	Querying MYSQL database	Write queries to interact with database tables to work around with data	K5
	Joining Tables	Apply various MySQL join clauses in the SELECT statement to retrieve data from two or more tables based on a related column between them.	K5
	Using logical operators	Access data in the database using more than one condition in WHERE clause using different logical operators.	K5
	MYSQL Functions	Categorize MYSQL functions and apply appropriately to acquire the enhanced capabilities of MYSQL	K3
	Accessing MYSQL via PHPMYADMIN	Establish Connection with MYSQL through the user friendly PHPMYADMIN which makes it easy to handle the database.	K6
IV	**Accessing MYSQL using PHP**		
	Querying MYSQL database through PHP	Use PHP to query MySQL database by entering the MySQL query command in the PHP script or define the command as a variable and use the variable when needed.	K5
	The $-POST array	Use $-POST to collect form data after submitting an HTML form with method="post".	K3
	Table Processing (Create, describe, drop,..)	Process tables in the database	K4
	Data Manipulation through PHP (Insert, update, select, delete)	Manipulate the stored data in the database through PHP based on the given problem.	K5
	Using AUTO¬_INCREMENT	Use MySQL's Auto increment feature to generate primary key	K3
	Performing additional queries	Develop the ability to perform additional queries other than common queries to access database.	K3
	Using Mysqli procedurally	Categorize the mysqli() functions and use to create queries for the given problem.	K3
	Form Handling: Building forms, Retrieving submitted data	Create forms to get user input and retrieve the submitted data for processing.	K6
	Register globals, Default values, input types, sanitizing input	Create forms with disabled register globals for security issues, set default values, and use appropriate	K6

		input types such as text boxes and text areas to checkboxes, radio buttons, and more for getting inputs, validate and sanitize user input.	
V	**Cookies, Sessions, Authentication and Validation**		
	Using cookies in PHP	Identify a user using cookie and track sessions.	K3
	HTTP Authentication	Manage users and passwords for the application by HTTP Authentication.	K4
	Using Sessions	Use Session variables (which hold information about one single user) to make data accessible across the various pages of an entire website.	K5
	Validating user Input	Create a form with user input validation	K4
	Regular Expression, Matching through metacharacters, fuzzy character matching, grouping	Accomplish pattern matching using regular expression, fuzzy character matching, and grouping.	K4
	Character classes, Indicating range, Negation - General Modifiers	Perform validation using Character classes, Indicating range, Negation - General Modifiers	K4
	Using regular expression in PHP	Apply regular expression functions preg_match, preg_match_all, and preg_replace for matching.	K5
	Redisplaying a form after PHP validation	Redisplay the created form after PHP validation.	K6

4. MAPPING SCHEME (CO, PO & PSO)

U21CS505	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2	PSO3	PSO4
CO1	H				H	H	H				H		H
CO2	H	L	M	M	H	M	H			H	H	H	
CO3	H	M	H	H	H		H			M			
CO4	H	H	H	H	H	M	H		H	H		H	H
CO5	H			M	H	M	H				H	H	M
CO6	H	H	H		H	H	H	H	H	H		H	H

5. COURSE ASSESSMENT METHOD

DIRECT

1. Continuous Assessment Test: T1, T2 (Theory & Practical Components): Closed Boo
2. Open Book Test.
3.Assignment, Group Discussion, Project Report, Seminar, Quiz (written).
4.Pre-Semester & End Semester Theory Examination

INDIRECT

1.Course end survey (Feedback)

Name of the Course Coordinator: Dr. Rama Sivakumar

CORE VI: OPERATING SYSTEM

SEMESTER: I **CODE: U21CS506**

CREDITS: 5 **HOURS /WEEK: 5**

----

1. COURSE OUTCOMES

CO #	Course Outcomes	Level	Unit
CO1	Examine the process and thread management for various OS	K5	1
CO2	Apply the concurrency management and deadlock avoidance in OS environment	K5	2
CO3	Assess the Memory management for different OS	K6	3
CO4	Evaluate the scheduling algorithms for OS	K6	4
CO5	Select the appropriate disk scheduling algorithm for OS	K5	5
CO6	Distinguish the strategies for file management in OS	K6	5

2. A. SYLLABUS

Unit I: Process and Threads

Operating System Overview: Operating System Objectives and Functions - Evolution of Operating Systems - Major Achievements - Developments Leading to Modern Operating Systems - Fault Tolerance - OS Design Considerations for Multiprocessor and Multicore - Microsoft Windows Overview - Traditional Unix Systems - Modern Unix Systems - Linux – Android. **Process Description and Control:** Process - Process States - Process Description - Process Control. **Threads:** Processes and Threads - Types of Threads - Multicore and Multithreading - Windows Process and Thread Management -Android Process and Thread Management.

Unit II: Concurrency and Deadlock

Concurrency: Mutual Exclusion: Software Approaches - Principles of Concurrency - Mutual Exclusion: Hardware Support – Semaphores. **Concurrency: Deadlock and Starvation:** Principles of Deadlock - Deadlock Prevention - Deadlock Avoidance - Deadlock Detection **-** An Integrated Deadlock Strategy - Dining Philosophers Problem - Windows Concurrency Mechanisms - Android Interprocess Communication.

Unit III: Memory Management

Memory Management: Memory Management Requirements - Memory Partitioning - Paging – Segmentation. **Virtual Memory:** Hardware and Control Structures - Operating System Software- Windows Memory Management - Android Memory Management.

Unit IV: Scheduling

Uniprocessor Scheduling: Types of Processor Scheduling - Scheduling Algorithms - Traditional Unix Scheduling - **Multiprocessor, Multicore, and Real-Time Scheduling:** Multiprocessor and Multicore Scheduling - Real-Time Scheduling - Linux Scheduling - Windows Scheduling.

Unit V: Input/Output And Files

Management and Disk Scheduling: I/O Devices - Organization of the I/O Function - Operating System Design Issues I/O Buffering - Disk Scheduling - Raid - Disk Cache – Windows I/O. **File Management:** Overview - File Organization and Access - B-Trees - File Directories - File Sharing - Record Blocking - Secondary Storage Management - Windows File System - Android File Management.

B. TOPICS FOR SELF STUDY

S.No.	Topics	Web Links
1	System Calls	https://youtu.be/lhToWeuWWfw
2	Socket in OS	https://youtu.be/uagKTbohimU
3	Shared Memory System	https://youtu.be/uHtzOFwgD74
4	Android Architecture	https://www.youtube.com/watch?v=y3HravY897E

C. TEXT BOOKS

1. William Stallings, " Operating Systems Internals and Design Principles", 9th Ed., Pearson, 2018

D. REFERENCES

1. Ida M. Flynn, Ann MClverMCHocs,*"Understanding Operating Systems, Course Technology"***,** 6th Edition, Cengage Learning,2011,.
2. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, *"Operating System Concepts"*, 9th Edition, Wiley Publication Inc, 2013.

E. WEB LINKS

- https://www.coursera.org/courses?query=operating%20system
- https://www.edx.org/learn/operating-systems
- https://www.tutorialspoint.com/operating_system/index.htm

3. SPECIFIC LEARNING OUTCOMES (SLO)

Unit/ Section	Course Content	Learning outcomes	Blooms Taxonomy Highest Level of Transaction
I	**Process and Threads**		
1.1	**Operating System Overview:** Operating System Objectives and Functions	Explain the features of Operating System	K5
1.2	Evolution of Operating Systems - Major Achievements	Explain the evolution of OS	K4
1.3	Developments Leading to Modern Operating Systems - Fault Tolerance - Considerations for Multiprocessor and Multicore OS Design	Evaluate the properties of fault tolerance, multiprocessor and multicore OS design	
1.4	Microsoft Windows Overview - Traditional Unix Systems - Modern Unix Systems - Linux – Android.	Compare the different types of Oss Assess the feature of Oss	K5 K5
1.5	**Process Description and Control:** Process - Process States - Process Description - Process Control.	Explain the process states Design the Process control	K5 K6
1.6	**Threads:** Processes and Threads - Types of Threads - Multicore and Multithreading	Explain the different types of thread Implement the multithreads in Multicore architecture	K4 K6

1.7	Windows Process and Thread Management -Android Process and Thread Management.	Apply the Process and Thread Management in different OS environment	K5
II	**Concurrency and Deadlock**		
2.1	**Concurrency:** Mutual Exclusion: Software Approaches	Develop the mutual exclusion for OS Explain the concepts of Mutual Exclusion	K6 K4
2.2	Principles of Concurrency - Mutual Exclusion: Hardware Support	Explain the mutual exclusion using Hardware Distinguish the software and hardware approaches of mutual exclusion	K4 K6
2.3	Semaphores	Design the semaphore for the OS	K6
2.4	**Deadlock and Starvation:** Principles of Deadlock	Explain the Deadlock and its causes	K4
2.5	Deadlock Prevention - Deadlock Avoidance	Develop the algorithms for deadlock prevention and avoidance	K6
2.6	Deadlock Detection - An Integrated Deadlock Strategy - Dining Philosophers Problem	Design the algorithm for deadlock detection Explain the role of dining philosophers in deadlock.	K6 K4
2.7	Windows Concurrency Mechanisms - Android Interprocess Communication	Apply the Concurrency and deadlock management for different types of OS	K5
III	**Memory Management**		
3.1	**Memory Management:** Memory Management Requirements	Assess the requirements for memory management	K6
3.2	Memory Partitioning	Build the strategies for memory partitioning	K6
3.3	Paging	Develop the algorithm for paging process	K6
3.4	Segmentation.	Construct the segmentation algorithms for OS	K6
3.5	**Virtual Memory:** Hardware and Control Structures - Operating System Software	Explain the virtual memory Assess the virtual memory using hardware and software implementation	
3.6	Windows Memory Management - Android Memory Management.	Apply the Memory management for Windows and Android	K6
IV	**Scheduling**		
4.1	**Uniprocessor Scheduling:** Types of Processor Scheduling	Explain the types of processor scheduling	K4
4.2	Scheduling Algorithms	Develop the scheduling algorithms for DBMS	K6
4.3	Traditional Unix Scheduling	Understand the scheduling in Unix	K5
4.4	**Multiprocessor, Multicore, and Real-Time Scheduling:** Multiprocessor and Multicore Scheduling	Explain the scheduling for Multiprocessor and Multicore	K4
4.5	Real-Time Scheduling	Design the Real-Time Scheduling	K6
4.6	Linux Scheduling - Windows Scheduling	Apply the scheduling in Linux and Windows	K4
V	**Input / Output And Files**		
5.1	**Management and Disk Scheduling:** I/O Devices - Organization of the I/O Function	Explain the I/O functions Assess the requirement of I/O devices	K4 K6

5.2	Operating System Design Issues I/O Buffering	Evaluate the solutions for Issues I/O Buffering	K5
5.3	Disk Scheduling - Raid - Disk Cache – Windows I/O.	Develop the Disk scheduling algorithms Compare the Raids in OS	K6 K6
5.4	**File Management:** Overview - File Organization and Access	Explain the file organizations Assess the file access procedures	K4 K5
5.5	B-Trees - File Directories - File Sharing - Record Blocking	Construct the B-trees for file organizations Evaluate the procedures for file sharing	K6 K5
5.6	Secondary Storage Management	Develop the secondary storage management	K6
5.7	Windows File System - Android File Management.	Appraise the files system of Windows and Android OS	K5

4. MAPPING SCHEME (CO, PO, PSO)

L-Low **M-Moderate** **H-High**

U21CS506	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2	PSO3	PSO4
CO1	H	-	M	M	L	-	M	M	-	H	-		M
CO2	H	-	M	M	M	-	-	-	-	H	-		M
CO3	H	-	M	M	M	-	M	M	-	H	-		M
CO4	H	-	M	M	M	-	-	-	-	H	-		H
CO5	H	M	M	M	M	-	-	-	-	H	M	M	H
CO6	H	H	H	H	H	M	-	-	-	H	M	M	H

5. COURSE ASSESSMENT METHODS

DIRECT:

1. Continuous Assessment Test: T1, T2 (Theory & Practical Components): Closed Book
2. Open Book Test, Assignment, Group Discussion, Seminar, Quiz.
3. Pre-Semester & End Semester Theory Examination

INDIRECT:

1. Course end survey (Feedback)

Name of the Course Coordinator: Dr. P. Thamil Selvan

CORE VII: FOUNDATION OF COMPUTER GRAPHICS

SEMESTER: V **CODE: U21CS507**

CREDITS: 5 **HOURS /WEEK: 5**

----

1. COURSE OUTCOMES

CO #	COURSE OUTCOMES	LEVEL	UNITS
CO1	Apply and Analyze the Scan conversion Algorithms for Point, Line and Circle.	K5	1
CO2	Evaluate the algorithms for 2D & 3D Transformations.	K6	2
CO3	Design the algorithms for 2D & 3DViewing and Clipping	K6	3
CO4	Judge the need to apply projections and the anomalies that exist	K5	3
CO5	Construct the algorithms for visible surface and computer animation	K6	4
CO6	Develop the programs for computer graphics using OpenGL	K6	5

2. A. SYLLABUS

Unit I: Computer Graphics & Rasterization

Introduction to Computer Graphics: Definition of Computer Graphics **-** Definition of Computer Aided Design (CAD) - Image Generation on Screen **-** Image Generating Techniques - Graphic User Interface (GUI) - Refresh Rate - Working of Laser Printers - **Vector Representation of Geometric Entities:** Introduction - Line Generation Algorithm Using Equation of Line - Line Generation Using DDA Algorithm - Bresenham's Line Generation Algorithm - Bresenham's Midpoint Circle Generation Algorithm - Bresenham's Midpoint Ellipse Generation Algorithm - Arc Generation Algorithm Using Trigonometric Function.

Unit II: Transformations

Two-Dimensional Transformation - Introduction - Representation of 2D Geometry -Types of 2D Transformations - Need of Homogeneous Coordinates - Special Transformation - Inverse Transformation. **Three-Dimensional Transformation:** Introduction - Scaling Transformation - Translation Transformation - Rotation Transformation - Reflection Transformation - Reflection About Any Arbitrary Plane in 3D Space - Shear Transformation.

Unit III: Windowing and Clipping & Projections

Windowing and Clipping : Introduction - Windowing - Clipping - Need for Windowing and Clipping - Viewing Transformation 2D Clipping - Cohen-Sutherland Subdivision Line Clipping Algorithm - Intersection Calculation and Clipping - Midpoint Subdivision Algorithm - Advantage of the Midpoint Subdivision Algorithm -Comparison Between Cohen-Sutherland and Midpoint Subdivision Line Clipping Algorithms - Polygon Clipping - Sutherland-Hodgman Algorithm - 3D Clipping - Multiple Windowing - Character Clipping - Applications of Clipping. **Projections:** Introduction - Projections - Perspective Projection - Parallel Projection - Differentials Between Parallel Projection and Perspective Projection.

Unit VI: Visible Surface Detection Methods & Computer Animation

Visible Surface Detection Methods: Classification of Visible-Surface - Detection Algorithms - Back-Face Detection -Depth-Buffer Method - A-Buffer Method - Scan-Line Method - Depth-Sorting Method - Area-Subdivision Method. **Computer Animation:** Raster Methods for Computer Animation - Design of Animation Sequences - Traditional Animation Techniques -General Computer-Animation Functions - Computer-Animation Languages - Key-Frame Systems - Motion Specifications - Character Animation - Periodic Motions.

Unit V: OpenGL

OpenGL with Computer Graphics: Introduction - Graphical Functions of OpenGL. **Programming Graphics Using OpenGL:** Application of Computer-Generated Images - Drawing Figures Using OpenGL - Drawing Tools - Transformation of Objects.

B. TOPICS FOR SELF STUDY

S.No.	Topics	Web Links
1	Introduction to LOGO language	https://www.tutorialspoint.com/logo/logo_quick_guide.htm
2	LOGO Primitives and Basic Commands	http://cs.brown.edu/people/orgs/artemis/old/2001/lessons/Logo.pdf
3	Computer Graphics and Animation	http://cs.wellesley.edu/~cs110/lectures/M01-color/graphics.pdf
4	Virtual Reality in computer graphics	https://web.mit.edu/16.459/www/VR1.pdf

C. TEXT BOOKS

1. D. P. Kothari, G. Awari, D. Shrimankar, A. Bhende, "Mathematics for Computer Graphics and Game Programming_ A Self-Teaching Introduction" , Mercury Learning & Information, 2019. **(Unit I, II, III, V)**

2. Donald D. Hearn/M. Pauline Baker, Warren Carithers, "Computer Graphics with Open GL", 4th Ed., Pearson, 2014. **(Unit IV)**

D. REFERENCES

1. ZHIGANG XIANG & ROY PLASTOCK, ***"Computer Graphics"*** Schaum's outline series McGraw-Hill International Edition, 2001.
2. Leen Ammeraal, Kang Zhang, "Computer Graphics for Java Programmers", Springer International Publishing, 2017.

E. WEB LINKS

- https://www.javatpoint.com/computer-graphics-tutorial
- https://www.tutorialspoint.com/computer_graphics/index.htm
- http://cs.wellesley.edu/~cs110/lectures/M01-color/graphics.pdf

3. SPECIFIC LEARNING OUTCOMES (SLO)

Unit/ Section	Course Content	Learning outcomes	Blooms Taxonomy Highest Level of Transaction
I	**Computer Graphics & Rasterization**		
1.1	**Introduction to Computer Graphics:** Definition of Computer Graphics - Definition of Computer Aided Design (CAD)	Explain the features of Computer Graphics Explain the CAD	K4 K4
1.2	Image Generation on Screen **-** Image Generating Techniques	Develop the techniques for Image Generation	K6
1.3	Graphic User Interface (GUI) - Refresh Rate - Working of Laser Printers	Discuss the GUI Define the Refresh rate Examine the working mechanism of Laser Printers	K2 K1 K5
1.4	**Vector Representation of Geometric Entities:** Introduction - Line Generation Algorithm Using Equation of	Develop the algorithm for Line Generation using OpenGL Analyze the potential of different line generation algorithm	

	Line - Line Generation Using DDA Algorithm - Bresenham's Line Generation Algorithm		
1.5	Bresenham's Midpoint Circle Generation Algorithm	Develop the algorithm for Circle generation	K6
1.6	Bresenham's Midpoint Ellipse Generation Algorithm	Explain the ellipse generation algorithms	K5
1.7	Arc Generation Algorithm Using Trigonometric Function.	Design the algorithm or arc generation	K6
II	**Transformations**		
2.1	**Two-Dimensional Transformation** - Introduction - Representation of 2D Geometry - Types of 2D Transformations.	Explain the Transformation on 2D domain Appraise the types of 2D Transformations	K5 K5
2.2	Need of Homogeneous Coordinates	Formulate the homogeneous coordinates for 2D transformations	
2.3	Special Transformation - Inverse Transformation	Design the special transformation and inverse transformation	K6
2.4	**Three-Dimensional Transformation:** Introduction - Scaling Transformation - Translation Transformation	Explain the features of transformations in 3D Develop the scaling transformation and translation Evaluate the scaling and translation transformation	K5 K6 K5
2.5	Rotation Transformation - Reflection Transformation	Apply the rotation and reflection on 3D	K5
2.6	Reflection About Any Arbitrary Plane in 3D Space	Explain the Reflection About Any Arbitrary Plane in 3D Space	K5
2.7	Shear Transformation	Design the shear transformation for 3D	K6
III	**Windowing and Clipping & Projections**		
3.1	**Windowing and Clipping :** Introduction - Windowing - Clipping - Need for Windowing and Clipping - Viewing Transformation	Explain the Windowing and clipping Assess the need for windowing and clipping Construct the viewing transformation	K4 K5 K6
3.2	2D Clipping - Cohen-Sutherland Subdivision Line Clipping Algorithm - Intersection Calculation and Clipping	Construct the 2D line clipping algorithms Compare and Asses the line clipping algorithms	K6 K4
3.3	Midpoint Subdivision Algorithm - Advantage of the Midpoint Subdivision Algorithm - Comparison Between Cohen-Sutherland and Midpoint Subdivision Line Clipping Algorithms	Construct the 2D line clipping algorithm using Midpoint subdivision Distinguish the Cohen Cohen-Sutherland and Midpoint Subdivision Line Clipping Algorithms	K6 K4
3.4	Polygon Clipping - Sutherland-Hodgman Algorithm	Develop the polygon clipping using Sutherland-Hodgman algorithm	K6
3.5	3D Clipping - Multiple Windowing - Character Clipping - Applications of Clipping	List the application of 3D clipping Analyze the multiple windowing strategy	K3 K4

3.6	**Projections:** Introduction - Projections - Perspective Projection - Parallel Projection -	Explain the role of projections in CG Explain the algorithm for parallel and perspective projection	K5 K5
3.7	Differentials Between Parallel Projection and Perspective Projection.	Analyze the features of Parallel Projection and Perspective Projection.	K4
IV	**Visible Surface Detection Methods & Computer Animation**		
4.1	**Visible Surface Detection Methods:** Classification of Visible-Surface.	Explain the visible surface Evaluate the classification visible surface.	K4 K5
4.2	Detection Algorithms - Back-Face Detection	Develop the detection algorithm using back face	K6
4.3	Depth-Buffer Method - A-Buffer Method	Design the algorithms using Buffer methods	K6
4.4	Scan-Line Method - Depth-Sorting Method - Area-Subdivision Method.	Formulate the methods for identifying the visible surface using scanline, depth sort and are subdivision	K6
4.5	**Computer Animation:** Raster Methods for Computer Animation - Design of Animation Sequences	Define the Computer Animation Evaluate the raster method for computer animation	K2 K5
4.6	Traditional Animation Techniques -General Computer-Animation Functions	Assess the traditional animation methods List the Computer animation functions	K4 K3
4.7	Computer-Animation Languages - Key-Frame Systems - Motion Specifications - Character Animation - Periodic Motions	Explain the key-frame system Evaluate the motion specifications in animation Develop the methods for character animation	K2 K5 K6
V	**OpenGL**		
5.1	**OpenGL with Computer Graphics:** Introduction - Graphical Functions of OpenGL. -	List the graphical functions of OpenGL Develop the programs using OpenGL	K2 K6
5.2	**Programming Graphics Using OpenGL:** Application of Computer-Generated Images	Write the OpenGL program computer generated images	K6
5.3	Drawing Figures Using OpenGL	Design the figures using OpenGL	K6
5.4	Drawing Tools	Assess the drawing tools in OpenGL	K5
5.5	Transformation of Objects.	Develop the program for transformation	K6

4. MAPPING SCHEME (CO, PO, PSO)

L-Low **M-Moderate** **H-High**

U21CS507	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2	PSO3	PSO4
CO1	H	-	M	M	L	-	M	M	-	H	-		M
CO2	H	-	M	M	M	-	-	-	-	H	-		M
CO3	H	-	M	M	M	-	M	M	-	H	-		M
CO4	H	-	M	M	M	-	-	-	-	H	-		H
CO5	H	M	M	M	M	-	-	-	-	H	M	M	H
CO6	H	H	H	H	H	M	-	-	-	H	M	M	H

5. COURSE ASSESSMENT METHODS

DIRECT:

1. Continuous Assessment Test: T1, T2 (Theory & Practical Components): Closed Book
2. Open Book Test, Assignment, Group Discussion, Seminar, Quiz.
3. End Semester Theory Examination

INDIRECT:

1. Course end survey (Feedback)

Name of the Course Coordinator : Dr. R. Cynthia Monica Priya

CORE PRACTICAL V: WEB PROGRAMMING LAB

SEMESTER: V **CODE: U21CS5P5**
CREDITS: 4 **HOURS /WEEK:6**

---

1. COURSE OUTCOMES

CO #	COURSE OUTCOMES	LEVEL	Ex. No.
CO1	Distinguish the language features of PHP including strings, regular expressions, files and forms.	K4	1-3
CO2	Experiment the OOP features to given problem scenarios.	K6	4-5
CO3	Select proper controls and built-in functions to create forms with image and file upload.	K5	6-7
CO4	Test MYSQL connection through PHPMYADMIN.	K6	8-9
CO5	Design web pages with Validation.	K6	10-14
CO6	Assemble the features of PHP and MYSQL and create a website.	K6	1-14

2. A. SYLLABUS

1. Construct a PHP program that converts given temperature in celcius to Farenheit and vice versa.
2. Develop an application to find the biggest of given three numbers using PHP.
3. Develop an application using String functions of PHP.
4. Develop an application to maintain student records using files.
5. Develop a simple calculator using PHP.
6. Develop a Login form using PHP.
7. Develop an application to display employee details submitted in forms.
8. Develop an application to display customer purchase details stored in MYSQL in table format.
9. Develop an application to manipulate student mark sheet stored in MYSQL via PHP.
10. Develop an application to upload a file that contains salary details of employees and check if the file already exists.
11. Develop a page counter application in PHP using Cookies.
12. Develop an application to track user sessions of an online store.
13. Develop a job application with image upload (photo) with required field validation.
14. Develop an application for digital library access with registration and login validation.

B. WEB LINKS

- https://www.youtube.com/watch?v=cGwSm8xDSwI&ab_channel=CodeWithDary
- https://www.youtube.com/watch?v=XBj_le81sAc&ab_channel=DaniKrossing
- https://www.youtube.com/watch?v=U10yvfIStx8&ab_channel=DaniKrossing

3. SPECIFIC LEARNING OUTCOMES

Exp. No	Title of the Experiment	K-level
1.	[Starter Applications]: Using echo and print statements Reversing the Number and string Palindrome Checking	K2
2.	[File Handling]: Displaying the contents of the file stored in the drive.	K3
3.	[Object Oriented Programming]: Demonstrating Constructors and Destructors by calculating the employee's salary.	K3
4.	[Looping and control Structures]: Generating Fibonacci Series of numbers Sorting the given numbers using for loop and if statement.	K3
5.	[Form Handling in PHP]: Creating a form Required field validation in form Validating form data using pattern matching	K5 K6
6.	[Cookies and Session in PHP]: Getting Customer Purchase details using cookies Counting the frequency of web page visit using session	K3
7.	[Image Upload]: Image Upload using PHP	K3
8.	[Database Connectivity– Create Table and Insert data]: Student Registration using PHP and MYSQL	K3 K6
9.	[Database Connectivity – View data]: Displaying employee details stored in database	K3 K6
10.	[Database Connectivity – formatting the data display]: Storing, Retrieving Customer purchase details and displaying in table format from database.	K3 K6

4. MAPPING SCHEME (CO, PO & PSO)

U21CS5P5	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2	PSO3	PSO4
CO1	H				H	H	H				H		H
CO2	H	L	M	M	H	M	H			H	H	H	
CO3	H	M	H	H	H		H			M			
CO4	H	H	H	H	H	M	H		H	H		H	H
CO5	H			M	H	M	H				H	H	M
CO6	H	H	H		H	H	H	H	H	H		H	H

5. COURSE ASSESSMENT METHODS

DIRECT:

1.Continuous Assessment Test: T1, T2 (Theory & Practical Components): Closed Book

2.Open Book Test.

3.Cooperative Learning Report, Assignment, Group Presentation, Seminar, Quiz (written).

4.Pre-Semester & End Semester Theory Examination

INDIRECT:

1.Course end survey (Feedback)

Name of the Course Coordinator : Dr. Rama Sivakumar

Elective - (1a) SOFTWARE ENGINEERING

SEMESTER: V **CODE: U21CS5:1**

CREDITS : 5 **HOURS /WEEK: 5**

----

1. COURSE OUTCOMES

CO #	COURSE OUTCOMES	LEVEL	UNIT
CO1	Define terms software, software engineering and the professional behavior.	K1	1
CO2	Experiment with various software process models for real world use-cases	K6	2
CO3	Evaluate the different types of software process models	K6	2
CO4	Develop software requirements specification.	K5	3
CO5	Apply design techniques and handle implementation issues.	K5	4
CO6	Compare various testing strategies and select appropriate one for testing software	K5	5

2. A. SYLLABUS

Unit I: INTRODUCTION TO SOFTWARE ENGINEERING 15

Building a software system: Characteristics of building a system. – Size, complexity, technical considerations – Building hypothetical system – Co-ordination efforts: Process, Product and people. – Engineering of software: Characteristics of software failures. – Software engineering definitions, Relevancy of software engineering and software. – Software engineering Profession and ethics: Code of ethics, Professional behavior – Principle of Software engineering: Davis Early Principle. – Principle of Software engineering: Royces More Principle – Principle of Software engineering: Wassermann's Fundamental Software engineering.

Unit II: SOFTWARE PROCESS MODELS 15

Software Process : Goal and the simplest process model. – Traditional Process Model: Water Fall Model. – Traditional Process Model: Chief Programmers Team Approach. – Traditional Process Model: Incremental Model. – Traditional Process Model: Spiral Model. – Entry and Exit Criteria: Entry Criteria. – Entry and Exit Criteria: Exit Criteria. – Process Assessment Models: SEI'S Capability Maturity Model. – Process Assessment Models: SEI'S Capability Maturity Model Integrated. – Process Definition and Communication.

Unit III: REQUIREMENTS ENGINEERING 15

Requirements Processing. – Requirements Elicitation and Gathering: Elicitation of high-level Requirements. – Elicitation of Detailed Requirements. – Requirements Analysis: By Business Flow. – View-Point Oriented Requirements Definitions. – Requirements Analysis and Prioritization. – Requirements Traceability. – Requirements Definitions, Prototyping and Reviews. – Requirements Specifications. – Requirements Agreements.

Unit IV: DESIGN AND IMPLEMENTATION 15

Introduction To Design: Architectural Design. – Introduction To Design: Detailed Design. – Design Characteristics. – Design Attributes. – Introduction to Implementation: Characteristics of good implementation. – Programming Style. – Coding Guidelines. – Comments. – Debugging. – Performance Optimization

Unit V: TESTING AND MAINTENANCE 15

Introduction To Testing: Testing and Quality Assurance – Purpose of Testing, Testing Techniques. – Inspections and Reviews. – Formal Methods. – Static Analysis. – Software Support and Maintenance. – Customer Support. – Product Maintenance Updates – Release Cycle. – Change Control.

B. TOPICS FOR SELF STUDY

S.No.	Topics	Web Links
1	Aspect Oriented programming	https://www.youtube.com/watch?v=DuFPj8MlAVo
2	Functional Decomposition	https://www.youtube.com/watch?v=XuDLUOzwBOs
3	Post Object Programming	https://www.youtube.com/watch?v=eFCI_F3D7WU
4	Software Maintainence	https://www.youtube.com/watch?v=8swQr0kckZI

C. TEXT BOOK

1. Frank Tsui, OrlandoKaram, Barbara Bernal **, *"Essentials of Software Engineering"***,4th Edition, Jones & Bartlett Learning, 2016.

D. REFERENCES BOOK

1. "Ian Somerville, *"**Software Engineering"***, 10th Edition, Pearson Edition,2015.

E. WEB LINKS

- https://www.edx.org/learn/software-engineering
- https://www.coursera.org/courses?query=software%20engineering
- https://www.mygreatlearning.com/software-engineering/courses

3. SPECIFIC LEARNING OUTCOMES (SLO)

Unit	Topics	Learning Outcomes	Level
I	**INTRODUCTION TO SOFTWARE ENGINEERING**		
1.1	Building a software system: Characteristics of building a system.	Defining the software and finding the characteristics of building a system	K1
1.2	Size, complexity, technical considerations – Building hypothetical system	Illustrate various factors to be considered for building a system	K2
1.3	Co-ordination efforts: Process, Product and people.	Relate the coordination efforts for software system	K2
1.4	Engineering of software: Characteristics of software failures - Software engineering definitions, Relevancy of software engineering and software	Outline the fundamentals of software engineering and explain its characteristics	K2
1.5	Software engineering Profession and ethics: Code of ethics, Professional behavior	Analyse different types of professional behaviour of a software engineer	K4
1.6	Principle of Software engineering: Davis Early Principle - Royces More Principle – Wassermann's Fundamental Software engineering.	Explain the principles of software engineering through familiar authors	K2
II	**SOFTWARE PROCESS MODELS**		
2.1	Software Process : Goal and the simplest process model	Illustrate the software process and models	K2
2.2	Traditional Process Model: Water Fall Model - Chief Programmers Team Approach - Incremental Model - Spiral Model	Compare various traditional process models and explain the fitness for various software development	K5
2.3	Entry and Exit Criteria	Interpret the criteria for entry and exit	K2

2.4	Process Assessment Models: SEI’S Capability Maturity Model. – SEI’S Capability Maturity Model Integrated	Demonstrate the process assessment models	K2
2.5	Process Definition and Communication.	Explain the process and the communication	K5
III	**REQUIREMENTS ENGINEERING**		
3.1	Requirements Processing	Interpret the requirements processing	K2
3.2	Requirements Elicitation and Gathering	Distinguish the high level and detailed elicitation and gathering	K4
3.3	Requirements Analysis	Analyse the requirements by business flow	K4
3.4	View-Point Oriented Requirements Definitions	Understanding the requirement gathering through different view points	K2
3.5	Requirements Analysis and Prioritization	Evaluate the requirements and prioritize the requirements	K5
3.6	Requirements Traceability	Interpret the traceability to track back the issues	K2
3.7	Requirements Definitions, Prototyping and Reviews	Explaining the central activities of requirements	K2
3.8	Requirements Specifications - Agreements	Compile the requirements and compose the specification	K4
IV	**DESIGN AND IMPLEMENTATION**		
4.1	Introduction To Design: Architectural Design – Detailed Design	Compare the two different designs	K5
4.2	Design Characteristics	Illustrate the characteristics of designing	K2
4.3	Design Attributes	Illustrate various attributes of designing	K2
4.4	Introduction to Implementation: Characteristics of good implementation	Identify various characteristics to be considered while development of a software	K3
4.5	Programming Style and Coding Guidelines	List out different styles and guidelines followed by organizations during software development	K4
4.6	Comments - Debugging	Explain the various ways of commenting and debugging	K5
4.7	Performance Optimization	Discuss optimizing the performance	K5
V	**TESTING AND MAINTENANCE**		
5.1	Testing, Quality Assurance and Purpose of Testing	Define testing and quality	K1
5.2	Testing Techniques	Demonstrate various techniques of testing	K2
5.3	Inspections and Reviews	Adapt the inspection and review process for testing	K6
5.4	Formal Methods	Interpret the mathematical techniques for testing	K2
5.5	Static Analysis	Apply the static analysis for detecting error-prone conditions	K3

5.6	Software Support and Maintenance - Customer Support	Understand various aspects of customer support after the product delivery	K2
5.7	Product Maintenance Updates	Demonstrate the maintenance of software and updating periodically	K4
5.8	Release Cycle – Change Control	Explain and summarize the releasing of software and outline the change control	K5

4. MAPPING SCHME (CO, PO & PSO)

U21CS5:1	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2	PSO3	PSO4
CO1	H	M	M	-	-	M	-	M	M	H	M	L	M
CO2	H	M	M	-	-	H	-	M	M	H	M	L	M
CO3	H	H	M	-	-	H	-	M	M	H	-	M	M
CO4	H	H	M	-	-	H	H	M	M	H	H	M	M
CO5	H	M	M	H	M	H	H	H	M	H	H	M	M
CO6	H	H	H	H	M	H	H	H	M	H	H	H	M

5. COURSE ASSESSMENT METHODS

DIRECT:

1. Continuous Assessment Test: T1, T2 (Theory & Practical Components): Closed Book
2. Open Book Test.
3. Cooperative Learning Report, Assignment, Group Presentation, Group Discussion, project Report, Field Visit Report, Poster Presentation, Seminar, Quiz (written).
4. Pre-Semester & End Semester Theory Examination

INDIRECT:

1. Course end survey (Feedback)

Name of the Course Coordinator: Dr. J. Isac Gnanaraj

SBEC – II: MINI PROJECT

SEMESTER: V **Code: U21CSPS2**
CREDITS : 2 **HOURS/WEEK : 2**

Course Outcomes

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

- Identify and define the problem statement
- Define and justify scope of the proposed problem
- Gather and analyze system requirements
- Propose an optimized solution among the existing solutions
- Practice software analysis and design techniques
- Develop technical report writing and oral presentation skills

Text Book(s):

1. Lynn E. Miner & Jeremy T. Miner, ***"Proposal Planning and Writing",*** Third Edition, Greenwood Publishing Group, 2003,

References Book(s):

1. William Navidi, ***"Statistics for Engineers and Scientists",*** 2nd Edition, McGraw-Hill, 2007.

SBEC – III: Technical Communication for Computer Scientists

SEMESTER: V **Code: U21CSPS3**
CREDITS : 2 **HOURS/WEEK : 2**

----

1. COUSE OUTCOMES

Upon completion of this course students should be able to

CO #	COURSE OUTCOMES	LEVEL	ACTIVITY NO.
CO1	Exhibit their communication with others through conversion	K5	1
CO2	Demonstrate their interview, discussion and marketing skills	K4	2
CO3	Analyze their reading ability	K5	3
CO4	Inculcate on their reading strategies	K5	4
CO5	Analyze their writing skills on note making, summarizing and essay	K5	5
CO6	Develop and design CV	K6	6

2. SYLLABUS

1. **Oral communication:** Starting and ending a conversation, telling and asking people to do things- expressing opinions and ideas- decisions and intentions- offers and invitations- feelings- right and wrong- numbers and money.
2. Purpose and audience- dealing with customers and clients- face to face discussions- meetings and attending meetings- checking understanding- raising questions- giving and receiving feedback- using body language- leading and directing discussions- concluding discussions, using graphics in oral presentation.
3. Reading comprehension and reference skills: Skimming and scanning, factual and inferential comprehension- prediction- guessing meaning of words from context- word reference- comprehending graphics in technical writing.
4. Reading strategies- reading speed, reading between lines for hidden meaning- interpreting graphics- using a dictionary- using an index- using a contents list to find information- choosing the right reference source.
5. **Written Communication:** Note making and note taking- summarizing- notes and memos- developing notes into text- organization of ideas- cohesion and coherence- paragraph writing- ordering information in space and time- short essays- description and argument- comparison and contrast- illustration- using graphics in writing- tables and charts- diagram and flow charts- maps, plans and graphs.
6. Spelling rules and tips- writing a rough draft- editing and proof reading- writing the final draft- styling text- filing in complex forms- standard letters- CV- writing a report- writing leaflets and brochures- writing references- essay writing- expository writing- description of processes and products- classification- the instructional process- arguments and presentation of arguments- narrating events chronologically.

CORE VIII: COMPUTER NETWORKING

SEMESTER: VI — **CODE :U21CS608**
CREDITS : 5 — **HOURS /WEEK: 6**

---

1. COURSE OUTCOMES

CO #	COURSE OUTCOMES	LEVEL	UNITS
CO1	Understand the fundamental concepts of computer networking.	K5	1
CO2	Analyze advanced networking concepts, preparing the student for entry Advanced courses in computer networking.	K5	2
CO3	Evaluate the OSI layers and working knowledge of datagram and internet socket programming.	K5	3
CO4	Explain network architecture using protocols and interfaces.	K5	4
CO5	Evaluate different encoding and decoding mechanisms involved in different types of transmission media and to measure the transmission impairments.	K6	5
CO6	Explain the different types of Link Layer Protocols	K5	5

2. A. SYLLABUS

Unit I: Introduction of Internet and Networking Protocols **15**
Internet (Network Description, Network Services & Protocol) - Components of a Computer Network : Access Network, Physical media - Network Switching : Packet Switching, Circuit Switching - Quantitative QoS(Delay, Loss, Throughput, and Bandwidth) : Overview of Delay and Packet Loss, Queuing Delay and Packet Loss, End-to-End Delay, Throughput in Computer Network - Protocols Layers and their Service Models : Layered Architecture, Encapsulation.

Unit II: Application Layer **15**
Network Application : Application Architectures - Process Communication - The web and HTTP : Overview of HTTP, HTTP Message format,Non persistent and Persistent Connection, Cookies, web caching-File Transfer Protocol (FTP) - Electronic Mail: SMTP, Mail Access Protocol - DNS : Services, Functions, Records and Messages - Peer-to-Peer Applications: File Distributions – Architectures, BitTorrent- Peer-to-Peer Applications : Distributed Hass Table- Socket Programming.

Unit III: Transport Layer **15**
Transport Layer Services - Multiplexing and De multiplexing (Connectionless & Connection-Oriented) - Connection Less Transport : UDP - Reliable Data Transfer - Connection-Oriented Transport: TCP Connection, TCP Segment Structure, Round-Trip Time Estimation, Reliable Data Transfer (Doubling the Time interval), Reliable data Transfer (fast Retransmit, Go-Back-N), TCP Connection Management - Congestion Control - TCP Congestion Control.

Unit IV: The Network Layer **15**
Responsibilities of Network Layer (Forwarding & Routing, Network Service Model) - Virtual Circuit and Datagram Network - Router: Input Processing, Switching, & Output Processing, Routing Control plane, Internet Protocol(IP): Datagram format - Internet Protocol(IP): IPv4 Addressing, ICMP, IPv6, & IP Security - Routing Algorithm: Distance Vector, Link State, Hierarchical - Routing in the Internet: RIP, OSPS, BGP - Broadcast and Multicast Routing.

Unit V: The Link Layer **15**
Link Layer: Services, Implementation - Error-Detection & Correction: Parity, Checksum, & CRC - Multi Access Link and Protocols: Channel Partitioning protocol, Talking-Turns Protocols DOCSIS, Random Access Protocol - Switched Local Area Network: Link-Layer Addressing and ARP, Ethernet, and Link Layer Switches - Link Virtualization - Data Center Networking.

B. TOPICS FOR SELF STUDY

S.No.	Topics	Web Links
1	Ease of trouble Shooting	https://www.youtube.com/watch?v=4KUHkBAaD5A
2	Ethical hacking	https://www.youtube.com/watch?v=dz7Ntp7KQGA
3	Cyber Security	https://www.youtube.com/watch?v=PlHnamdwGmw
4	Link Virtualization	https://www.youtube.com/watch?v=3lAJWnAQlhk

C. TEXT BOOK

1. James F Kurose and Keith W. Ross, ***"Computer Networking"*** – A Top-Down Approach, Sixth Edition, 2017, Pearson Publication.

D. REFERENCES BOOK

1. Mani Subramanian; Tlmothy A. Gonsalves; N. Usha Rani, **"*Network Management: Principles and Practice"***, Pearson Education India, 2010

E. WEB LINKS

- https://www.coursera.org/courses?query=computer%20network
- https://www.edx.org/learn/computer-networking
- https://www.udacity.com/course/computer-networking--ud436

3. SPECIFIC LEARNING OUTCOMES

Unit	Learning Outcomes		Level
I	**Introduction of Internet and Networking Protocols**		
1.1	Internet , Components of a Computer	Provides an internet and network services with real life examples	K1
1.2	Network, Network Switching	1. Recall different types of network access. 2.Provides managing network in an Organization	K1
1.3	Quantitative QoS	Understand Quality of Service with delay, loss and bandwidth.	K2
	Protocols Layers and their service models	Illustrate a protocol and test its efficiency with different layers	K2
II	**Application Layer**		
2.1	Network Application -	Explain network architecture using protocols and interfaces.	K5
2.2	Process Communication	Understand the mechanisms of operating system to handle process and threads and their communications.	K2
2.3	The web and HTTP	Identify the Delivered data (HTML files, image files, query results, etc.) on the World Wide Web	K3
2.4	File Transfer Protocol	1. Apply the internet-based applications using programs on their computer to access and store emails	K3

		2. Operate through the reliable connection as a transfer session between the client and server computers..	
2.5	Electronic Mail, DNS	Find the solutions for network directions on the internet used to resolve host names	K1
2.6	File Distributions, Distributed Hass Table, Socket Programming	1. Illustrate the schema of distribution files. 2. Explain Communication between various processes usually running on different systems using client-server environment. 3.Create the server and client program and an example program.	K6
III	**Transport Layer**		
	Transport Layer Services , Multiplexing and De multiplexing, Connection Less Transport	1. Learn the process of multiple data streams from different resources. 2. Illustrate the length of time it takes for a signal to be sent plus the length of time it takes for an acknowledgement of that signal to be received	K2
	Reliable Data Transfer, TCP Connection, Congestion Control	1. Summarize delivery of all packets and to enable the receiver to deliver the packets in order to its application layer. 2. Explain the use of process of regulating the total amount of data entering the network	K2
IV	**The Network Layer**		
4.1	Responsibilities of Network Layer, Virtual Circuit and Datagram Network , Router	1. Apply different encoding and decoding mechanisms involved. 2. Recall different types of transmission media and to measure the transmission impairments. 3. Learn the order of packet sending.	K3
4.2	Internet Protocol - Routing Algorithm - Routing in the Internet - Broadcast and Multicast Routing	1. Illustrate the principal set of digital message formats and rules for exchanging messages between computers across a single network or a series of interconnected networks. 2. Define set of step-by-step operations used to direct Internet traffic efficiently. 3. Recall broadcasting and multicasting process	K2
V	**The Link Layer**		
5.1	Link Layer - Error-Detection & Correction - Multi Access Link and Protocols	1. Analyse a model internet with various categories of networks and test the transmission rate. 2. Recollect the detection of errors caused by noise or other impairments during transmission from the transmitter to the receiver. 3. Define transmitting node always transmits at the full rate of the channel, namely, R bps using random access protocol	K4
5.2	Switched Local Area Network - Link Virtualization - Data Center Networking	1. Understanding Local Area Networking Learning Objectives 2. Learn on what is a Client and a Server 3. Summarize the Switches operations in the data link layer and sometimes the network	K2

		layer of the OSI Reference Model and therefore support any packet protocol	

4. MAPPING SCHEME (CO, PO & PSO)

L-Low M-Moderate H-High

U21CS608	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2	PSO3	PSO4
CO1	H	M	M	-	-	M	-	M	M	M	M	L	L
CO2	H	M	M	-	-	H	-	M	M	M	M	L	L
CO3	H	H	M	-	-	H	-	M	M	L	-	M	L
CO4	H	H	M	-	-	H	H	M	M	L	H	M	L
CO5	H	M	M	H	M	H	H	H	M	M	H	M	L
CO6	H	H	H	H	M	H	H	H	M	M	H	H	M

5. COURSE ASSESSMENT METHODS

DIRECT:

1. Continuous Assessment Test: T1, T2 (Theory & Practical Components): Closed Book
2. Open Book Test, Assignment, Group Discussion, Seminar, Quiz.
3. End Semester Theory Examination

INDIRECT:

1.Course end survey (Feedback)

Name of the Course Coordinator: Dr. B. Karthikeyan

ELECTIVE 2: MACHINE LEARNING

SEMESTER: VI **CODE: U21CS6:1**

CREDITS: 5 **HOURS /WEEK: 5**

---

1. COURSE OUTCOMES

CO #	COURSE OUTCOMES	LEVEL	UNITS
CO1	Apply and Analyze the Scan conversion Algorithms for Point, Line and Circle.	K5	1
CO2	Evaluate the algorithms for 2D & 3D Transformations.	K6	2
CO3	Design the algorithms for 2D & 3DViewing and Clipping	K6	3
CO4	Judge the need to apply projections and the anomalies that exist	K5	3
CO5	Construct the algorithms for visible surface and computer animation	K6	4
CO6	Develop the programs for computer graphics using OpenGL	K6	5

2. A. SYLLABUS

Unit I: Machine Learning Basics

Machine Learning Basics- Supervised learning - Unsupervised learning - Semi-supervised learning - Function approximation. **Data Preprocessing:** Feature extraction – Sampling - Data transformation - Outlier removal - Data deduplication - Relevance filtering - Normalization, discretization and aggregation - Entity resolution.

Unit II: Supervised Learning

Supervised learning: Classification - Regression analysis - Logistic regression - Evaluation of learner - Evaluating a learner

Unit III: Unsupervised and Semi-Supervised Learning

Unsupervised learning: Types of clustering - k-means clustering - Hierarchical clustering - Visualizing clusters - Evaluation of clusters. Semi-supervised learning: Expectation maximization - Pseudo labeling.

Unit IV: Learning Techniques

Learning Techniques: Learning techniques - Learning issues - Cross-validation - Ensemble learning - Reinforcement learning - Active learning - Machine teaching - Automated machine learning.

Unit V: Machine Learning Applications

Anomaly detection - Security – Predictive Maintenance - Biomedical applications – Medical Applications - Natural language processing – Text Mining - Other applications

B. TOPICS FOR SELF STUDY

S.No.	Topics	Web Links
1	Pytorch	https://pytorch.org/
2	TensorFlow	https://www.tensorflow.org/
3	Scikit Learn	https://scikit-learn.org/

C. TEXT BOOKS

1. Peter Wlodarczak, “ Machine Learning and its Applications”, 1st Ed., CRC Press, 2020.

D. REFERENCES

1. Kevin P. Murphy, "Machine Learning: A Probabilistic Perspective" by The MIT Press, 2012.

2. Ethem Alpaydin, "Introduction to Machine Learning" 2nd Edition, The MIT Press, 2009.
3. Tom M. Mitchell, "Machine Learning", First Edition by Tata McGraw-Hill Education, 2013.
4. Christopher M. Bishop, "Pattern Recognition and Machine Learning" by Springer, 2007.

E. WEB LINKS

- https://www.coursera.org/learn/machine-learning
- https://developers.google.com/machine-learning/crash-course
- https://machinelearningmastery.com/self-study-guide-to-machine-learning/#

3. SPECIFIC LEARNING OUTCOMES (SLO)

Unit/ Section	Course Content	Learning outcomes	Blooms Taxonomy Highest Level of Transaction
I	**Machine Learning Basics**		
1.1	Machine Learning Basics	Explain the basics of Machine Learning	K4
1.2	Supervised learning	Define the supervised learning	K2
1.3	Unsupervised learning - Semi-supervised learning	Understand the unsupervised and semi-supervised algorithm	K2
1.4	Function approximation	Construct the function approximation	K6
1.5	Data Preprocessing: Feature extraction – Sampling - Data transformation	Assess the data preprocessing methods Explain the feature extraction	K5 K4
1.6	Outlier removal - Data deduplication - Relevance filtering	Develop the algorithm for outlier removal Design the relevance filter Define data deduplication	K6 K6 K2
1.7	Normalization, discretization and aggregation - Entity resolution	List the normalization methods Explain the entity resolution	K2 K5
II	**Supervised Learning**		
2.1	Supervised learning	Describe the supervised learning	K4
2.2	Classification	Develop the algorithms for classification	K6
2.3	Regression analysis	Design the regression methods for the data	K6
2.4	Logistic regression	Explain the features of logistic regression	K4
2.5	Evaluation of learner - Evaluating a learner	Evaluate the learner techniques	K5
III	**Unsupervised and Semi-Supervised Learning**		
3.1	Unsupervised learning	Explain the unsupervised learning	K5
3.2	Types of clustering	List the types of clustering	K4
3.3	k-means clustering	Design the k-means clustering for data	K6
3.4	Hierarchical clustering	Build the hierarchical clustering for data	K6
3.5	Visualizing clusters	Analyze he clustering using visualization	K5
3.6	Evaluation of clusters	Assess the clusters of the data	K5
3.7	Semi-supervised learning: Expectation maximization	Develop the Expectation maximization for the data	K6
3.8	Pseudo labeling	Explain the pseudo labelling	K5
IV	**Learning Techniques**		

4.1	Learning techniques - Learning issues	Explain the learning techniques Assess the issues related to learning	K4 K5
4.2	Cross-validation	Construct the cross validation methods	K6
4.3	Ensemble learning	Design the ensemble learning	K6
4.4	Reinforcement learning	Develop the reinforcement learning	K6
4.5	Active learning	Build the active learning	K6
4.6	Machine teaching	Explain the machine teaching	K5
4.7	Automated machine learning.	Construct the automated machine learning	K6
V	**Machine Learning Applications**		
5.1	Anomaly detection	Develop the anomaly detection algorithms	K6
5.2	Security – Predictive Maintenance	Explain the role of machine learning in security and predictive maintenance	K5
5.3	Biomedicale applications	Design the biomedical application using ML	K6
5.4	Medical Applications	Design the medical applications using ML	K6
5.5	Natural language processing –	Build the NLP algorithm using ML	K6
5.6	Text Mining	Apply the text mining using ML	K4
5.7	Other applications	Analyze the other applications of ML	K4

4. MAPPING SCHEME (CO, PO, PSO)

L-Low M-Moderate H-High

U21CS6:1	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2	PSO3	PSO4
CO1	H	-	M	M	L	-	M	M	-	H	-		M
CO2	H	-	M	M	M	-	-	-	-	H	-		M
CO3	H	-	M	M	M	-	M	M	-	H	-		M
CO4	H	-	M	M	M	-	-	-	-	H	-		H
CO5	H	M	M	M	M	-	-	-	-	H	M	M	H
CO6	H	H	H	H	H	M	-	-	-	H	M	M	H

5. COURSE ASSESSMENT METHODS

DIRECT:

1. Continuous Assessment Test: T1, T2 (Theory & Practical Components)
2. Open Book Test, Assignment, Group Discussion, Seminar, Quiz
3. End Semester Theory Examination

INDIRECT:

1. Course end survey (Feedback)

Name of the Course Coordinator : Dr. P. S. Eliahim Jeevaraj

Elective – (3a) BUSINESS ANALYTICS

SEMESTER: VI **CODE: U18CS6:4**

CREDITS : 5 **HOURS /WEEK: 6**

----

1. COURSE OUTCOMES

CO #	COURSE OUTCOMES	LEVEL	UNITS
CO1	Apply various Excel tools and add-ins for analyzing Business problems.	K5	1
CO2	Compare mathematical formulas with Spreadsheet formulas	K5	2
CO3	Explore, query and summarize business data.	K6	3
CO4	Apply descriptive statistical measures for business decision.	K6	4
CO5	Perform progression analysis and forecasting techniques.	K6	5
CO6	Create trend lines using excel	K6	5

2. A. SYLLABUS

Unit I: INTRODUCTION TO BUSINESS ANALYTICS **15**

What is Business Analytics?- Evolution of Business Analytics-Scope of Business Analytics- Data for Business Analytics-Models in Business Analytics-Problem solving with Analytics.

Unit II: ANALYTICS ON SPREADSHEETS **15**

Basic Excel skills - Basic Excel Functions - Using Excel Lookup functions for Database Queries - Spreadsheet Add-Ins for Business Analytics.

Unit III: DESCRIPTIVE ANALYTICS – Visualizing and Exploring Data **15**

Data Visualization - Creating charts in Microsoft Excel - Other Excel visualization tools - Data queries: Tables, Sorting and Filtering – Statistical Methods for Summarizing Data.

Unit IV: DESCRIPTIVE STATISTICAL MEASURES **15**

Population and Samples - Measures of Location - Measures of Dispersion - Measures of Association - Measures of Shape - Excel Descriptive Statistical Tool - Statistical thinking in Business Decisions.

Unit V: PREDICTIVE ANALYTICS –Trend lines and Regression Analysis **15**

Trend Lines and Regression Analysis: Modeling Relationships and trends in data- Simple Linear Regression- Forecasting Techniques: Qualitative and Judgmental forecasting-Historical Analogy – The Delphi Method – Statistical Forecasting models – Forecasting models for stationary time series.

B. TOPICS FOR SELF STUDY

S.No.	Topics	Web Links
1	Data Vs Business Analytics	https://www.youtube.com/watch?v=AodyW7bhku8
2	Business analytical Skills	https://www.youtube.com/watch?v=xybB1tISxpk
3	Crypto Currency	https://www.youtube.com/watch?v=8NgVGnX4KOw
4	Forecasting Methods	https://www.youtube.com/watch?v=fp-1_9mLlbc

C. TEXT BOOK

1. ***"Business Analytics"***, James R. Evans, Second Edition, Pearson Education, 2016. Indian Edition 2017, Pearson India Services.

D. REFERENCES BOOK

1. ***"Essentials of Business Analytics"***, CAMM, COCHRAN, FRY, OHLMANN, ANDERSON, SWEENEY, WILLIAMS, 2015, CENGAGE LEARNING

E. WEB LINKS

- https://www.udemy.com/course/excel-for-business-analysts-online-course/
- https://www.simplilearn.com/learn-business-analytics-excel-fundamentals-skillup
- https://www.coursera.org/courses?query=data%20analysis%20excel

3. SPECIFIC LEARNING OUTCOMES (SLO)

Unit/ Section	Course Content	Learning outcomes	Blooms Taxonomy Level of Transaction
I	INTRODUCTION TO BUSINESS ANALYTICS		
1.1	What is Business Analytics?- Data for Business Analytics	Detailed explanation of Business analytics and scope	K3-K4
1.2	Evolution of Business Analytics	Description of Business Analytics	K2
1.3	Scope of Business Analytics	Lists the scope of Business Analytics	K1
1.4	Models in Business Analytics-Problem solving with Analytics.	Explaining the models of Business Analytics	K4-K5
II	ANALYTICS ON SPREADSHEETS		
2.1	Basic Excel skills	Defines the basic Excel skills and functions	K3
2.2	Basic Functions of Excel	Recall basic functions of Excel	K1
2.3	Using Excel Lookup functions for Database Queries	Construct database Queries	K4-K5
2.4	Build spreadsheet Add-Ins for Business Analytics	Spreadsheet Add-Ins for Business Analytics.	K5
III	DESCRIPTIVE ANALYTICS		
3.1	What is data visualization	Visualizing Data	K1
3.2	Illustrate the data charts in MS Excel	Creating charts in Microsoft Excel	K4
3.3	Propose Excel visualization tools	Analyzing and exploring the data's of Excel using different methods and tools	K5-K6
3.4	Construct data queries	Data queries from tables	K3
3.5	Show sorting and filtering of data queries	Sorting and filtering	K2

3.6	Analysis of statistical method for summarized data	Describes the various aspects and the functions of the Exce	K4
IV	**DESCRIPTIVE STATISTICAL MEASURES**		
4.1	Interpret Population and Samples	Measuring the Population	K2
4.2	**How to measure location**	Measures of Location	K1
4.3	**Evaluate measures of Dispersion**	Measuring the Population with different samples and measures	K5
4.4	**Originate the Measures of Association**	Explain the measures of Association	K6
4.5	**Measures of Shape - Excel Descriptive Statistical Tool**	Illustrating the measurements of the Statistical dispersion	K4
4.6	**Analysis of Statistical thinking in Business Decisions.**	Statistical dispersion	K4-K5
V	**PREDICTIVE ANALYTICS**		
5.1	Trend lines and Regression Analysis, Modeling Relationships and trends in data- Simple Linear Regression	Predicting the data's using analytical methods	K5
5.2	Originate Qualitative and Judgmental forecasting	Forecasting Techniques: Qualitative and Judgmental forecasting	K6
5.3	Recall Historical Analogy	Describes about the Trend Lines and the forecasting models.	K1
5.4	Summarize Delphi Method	Elaborating Delphi method	K2
5.5	Statistical Forecasting models – Forecasting models for stationary time series.	Distinguish Various Forecasting models	K4

4. MAPPING SCHME (CO, PO & PSO)

U21CS6:4	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PSO1	PSO2	PSO3	PSO4
CO1	H	M	M	-	-	M	-	M	M	M	M	L	L
CO2	H	M	M	-	-	H	-	M	M	M	M	L	L
CO3	H	H	M	-	-	H	-	M	M	L	-	M	L
CO4	H	H	M	-	-	H	H	M	M	L	H	M	L
CO5	H	M	M	H	M	H	H	H	M	M	H	M	L
CO6	H	H	H	H	M	H	H	H	M	M	H	H	M

5. COURSE ASSESSMENT METHODS

DIRECT:

1. Continuous Assessment Test: T1, T2 (Theory & Practical Components)
2. Open Book Test, Assignment, Group Discussion, Seminar, Quiz
3. End Semester Theory Examination

INDIRECT:

1. Course end survey (Feedback)

Name of the Course Coordinator: Dr. D. Suresh Kumar

PROJECT IMPLEMENTATION

SEMESTER: VI **Code: U21CS6PJ**
CREDITS : 5 **HOURS/WEEK : 6**

Course Outcomes

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

- Develop a functional application based on the software design
- Apply coding, debugging and testing tools to enhance the quality of the software
- Construct new software system based on the theory and practice gained through this exercise
- Prepare the proper documentation of software projects following the standard guidelines
- Learn technical report and oral presentation skills

Text Book(s):

1. Lynn E. Miner & Jeremy T. Miner, ***"Proposal Planning and Writing",*** Third Edition, Greenwood Publishing Group, 2003.

References Book(s):

1. William Navidi, ***"Statistics for Engineers and Scientists",*** 2nd Edition, McGraw-Hill, 2007.