BASAVESHWAR ENGINEERING COLLEGE (AUTONOMOUS), BAGALKOT

COURSE PLAN

Title of Course	:	**Object Oriented Programming with Java**	**Course Code**	:	**22UCS302C**
Credits	:	**03**	**Contact Hours/ Week**	:	**(2+2)**
Total Hours	:	**40**	**Tutorial Hours**	:	**0**
CIE Marks	:	**50**	**SEE Marks**	:	**50**
Semester	:	**3**	**Year**	:	**2023-24**

Prerequisites:

Nil

Course Objectives:

	The Course objectives are:
1	Learn object oriented concepts using programming examples
2	Study the concepts of importing of packages and exception handling mechanism
3	Study the concepts of Multithreading
4	Study the concepts of JDBC

Course Outcomes:

	At the end of the course, the student should be able to:
1	Explain the object-oriented concepts and other features of JAVA
2	Demonstrate the concepts of packages, interfaces, exception handling
3	Demonstrate the concepts of multithreading and Lambda expressions
4	Demonstrate the database access and details for managing information using the JDBC API
5	Design and develop solutions to real time problems using the JAVA.

Course Articulation Matrix: Mapping of Course Outcomes (CO) with Programme Outcomes (PO) and Programme Specific Outcomes (PSO)

		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
No	Programme Outcomes / Course Outcomes															
The students will be able to:																
1	Explain the object-oriented concepts and other features of JAVA	3	2											1		
2	Demonstrate the concepts of packages, interfaces, exception handling	1	2											2		1
3	Demonstrate the concepts of multithreading and Lambda expressions	1	3											1		2
4	Demonstrate the database access and details for managing information using the JDBC API	2	2											1		1
5	Design and develop solutions to real time problems using the JAVA.		3											2		1

Competencies Addressed in the course and Corresponding Performance Indicators

Programme Outcome:Any of 1 to 12 PO's:

PO 1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization for the solution of complex engineering problems.	
Competency	**Indicators**
1.4 Demonstrate competencein specialized engineering knowledge othe program	1.4.1Applytheoryandprinciplesofcomputerscienceandengineeringtosolveanengineeringproblem
PO2: Problem analysis: Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	
Competency	**Indicators**

2.1 Demonstrateanabilityto identify and formulatecomplexengineeringproblem	2.1.1 Identifyprocesses/modules/algorithmsofacomputer-basedsystemandparameterstosolveaproblem
PO 3: Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.	
Competency	**Indicators**
3.8Demonstrate an ability toadvanceanengineeringdesigntodefinedendstate	3.8.2 Abletoimplementandintegratethemodules 3.8.3 Able toverifythefunctionalitiesandvalidatethedesign.

PO1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

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

PO9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Unit Learning Outcomes (ULO):

Unit Learning Outcome (ULO)	CO	BLL	PI addressed
Explain the basic of Object Oriented Programming	**1**	**1,2**	**1.4.1**
Apply the concept of packages, interfaces , exception handling	**2**	**3**	**1.4.1,2.1.1**
Apply the concept of Multithreading and Lambda based expression	**3**	**3**	**1.4.1,2.1.1**
Apply the concept of JDBC	**4**	**3,4**	**1.4.1,**2.1.1
Design and Develop Solutions to real time problems using JAVA	**5**	**4**	**1.4.1,2.1.1,** 3.8.2, 3.8.3

Course Content:

Hours Required	Topic to be covered	Mode of Delivery
01	An overview of Java, Data Types, Variables	i. Chalk and talk in classroom ii. Lecture combined with discussions iii. Assignments
01	Arrays	
01	Control statements	
01	Introducing Classes: Class Fundamentals, Declaring Objects	
01	Constructor, parameterized constructor	
01	Garbage Collection	
01	Method overloading	
01	String Handling	
01	Inheritance: basic concepts	
01	Types of inheritance	
01	Packages	
01	Interfaces	
01	Exception-Handling Fundamentals	
01	Types of exception handling	
01	Uncaught Exceptions, Using try and catch	
01	Multiple catch clauses	
01	Lambda Expressions: Fundamentals	
01	Block Lambda expressions, Passing Lambda Expressions as argument	
01	Lambda Expressions and Exceptions.	
01	Multithreaded Programming: The Java Thread Model, The Main Thread	
01	Creating a Thread, Creating Multiple Threads	
01	Thread Priorities	
01	Synchronization	
01	Inter thread communication	
01	The Concept of JDBC	
01	JDBC Driver Types	
01	JDBC Packages, A Brief Overview of the JDBC process	
01	Database Connection	

01	Associating the JDBC/ODBC Bridge with the Database	
01	Statement object	
01	Result Set Objects	
01	Transaction Processing	

Review Questions:

Review Questions	ULO	BLL	PI addressed
How does Java achieve platform independence?	1	1	**1.4.1**
What is the purpose of garbage collection in Java, and when is it used	1	2	**1.4.1**
With syntax and examples, initialization of one-dimensional and two-dimensional arrays.	1	3	**1.4.1**
What is the usage of super keyword	2	2	**1.4.1**
How are exceptions handled in Java	2	2	**1.4.1**
What is the ideal usage of Lamda expression	3	2	**1.4.1**
Describe synchronization with respect to multithreading.	3	2	**2.1.1**
Write a class that implements the Char Sequence interface found in the java.lang package. Implementation should return the string backwards. Select one of the sentences from this book to use as the data.	3	3	**1.4.1**
Analyse the different ways of using thread	3	3	**2.1.1**
Types of JDBC driver	2	2	**1.4.1**
Explain the JDBC API components	2	2	**2.1.1**

Evaluation Scheme:

Assessment	Marks	Weightage
CIE-I	20	20
CIE-II	20	20
Assignments/ Quizzes/Case Study/ Course Project/Term Paper/Field Work	10	10
SEE	50	50
Total	100	100

Details of Assignment:

Assignment	Marks (10)	CO	PI	CA	PO
Assignment I :Writeup to explain or solve problems	5	1,2,3,4,5	1.4.1	--	1,2,3
Assignment 2:MCQ based QUIZ	5	2,3,4	1.4.1	--	1,2,3

BASAVESHWAR ENGINEERING COLLEGE (AUTONOMOUS), BAGALKOT

MODEL COURSEPLAN

Title of Course	**:**	**Computer Organization**	**Course Code**	**:**	**22UCS303C**
Credits	**:**	**3**	**Contact Hours/ Week**	**:**	**3**
Total Hours	**:**	**40**	**Tutorial Hours**	**:**	**--**
CIE Marks	**:**	**50**	**SEE Marks**	**:**	**50**
Semester	**:**	**III**	**Year**	**:**	**2023-2024**

Prerequisites:

Course Objectives:

	The Course objectives are:
1	To understand the design and interaction concepts of modern computer subsystem
2	To learn the technique for analyzing the performance of computer system.

Course Outcomes:

	At the end of the course the student should be able to:
1	Explain the design and function of different units of computer
2	Perform the various operations on given data
3	Analyse the execution of the program and different organizations of functional units
4	Develop an assembly programs and micro programs for simple machine instructions
5	Design the basic functional units of computer

Course Articulation Matrix: Mapping of Course Outcomes (CO) with Programme Outcomes (PO) and Programme Specific Outcomes (PSO)

		PO 1	PO2	PO3	PO4	PO5	PO6	PO 7	PO8	PO9	PO 10	PO 11	PO 12	PSO1	PSO2	PSO3
No	**Programme Outcomes / Course Outcomes**															
The students will be able to:																
1	Explain the design and function of different units of Computer	1											1	1		
2	Perform the various operations on given data		3										1	1		2
3	Analyse the execution of the program and different organizations of functional units		2	2									1	1		2
4	Develop assembly programs and micro programs for simple machine instructions			3									1	1		2
5	Design the basic functional units of computer			3									1	1		3

Competencies Addressed in the course and Corresponding Performance Indicators

Programme Outcome:Any of 1 to 12 PO's:

PO 1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialisation for the solution of complex engineering problems.	
Competency	**Indicators**
1.6 Demonstrate competence in engineering fundamentals	1.6.1 Apply engineering fundamentals
1.7 Demonstrate competence in specialized engineering knowledge to the program	1.7.1 Apply theory and principles of computer science and engineering to solve an engineering problem
PO2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	
Competency	**Indicators**

2.8 Demonstrate an ability to execute a solution process and analyze results	2.8.1 Applies engineering mathematics to implement the solution. 2.8.2 Analyze and interpret the results using contemporary tools.
PO 3: Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.	
Competency	**Indicators**
3.6 Demonstrate an ability to generate a diverse set of alternative design solutions	3.6.1 Able to explore design alternatives. 3.6.2 Able to produce a variety of potential design solutions suited to meet functional requirements.
3.7 Demonstrate an ability to select optimal design scheme for further development	3.7.1 Able to perform systematic evaluation of the degree to which several design concepts meet the criteria.
3.8 Demonstrate an ability to advance an engineering design to defined end state	3.8.1 Able to refine architecture design into a detailed design within the existing constraints. 3.8.2 Able to implement and integrate the modules. 3.8.3 Able to verify the functionalities and validate the design.
PO 12: Life-long learning: Recognise the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.	
Competency	**Indicators**
12.4 Demonstrate an ability to identify gaps in knowledge and a strategy to close these gaps	12.4.1 Describe the rationale for the requirement for continuing professional development 12.4.2 Identify deficiencies or gaps in knowledge and demonstrate an ability to source information to close this gap
12.5 Demonstrate an ability to identify changing trends in engineering knowledge and practice	12.5.1 Identify historic points of technological advance in engineering that required practitioners to seek education in order to stay current 12.5.2 Recognize the need and be able to clearly explain why it is vitally important to keep current regarding new developments in your field

PO1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

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

PO9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PSO1: Inculcate analytical, logical, programming and managerial abilities.

PSO2: Apply Software Engineering Principles and Practices to provide software solutions

PSO3: Design and Develop computational systems for real-life applications.

Unit Learning Outcomes (ULO):

Unit 1 Learning Outcome (ULO)	CO	BLL	PI addressed
Explain the functioning of different components of the computer system.	1	2	1.6.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Write assembly programs	4	3	3.7.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Analyze the execution of the program	2	4	2.8.1, 2.8.2,12.4.1, 12.4.2 12.5.1, 12.5.2

Unit 2 Learning Outcome (ULO)	CO	BLL	PI addressed
Explain the functioning of different components of the computer system.	1	2	1.6,12.4, 12.5
Write assembly programs	4	3	3.7.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Design basic I/O interface circuits	5	3	3.7.1,12.4.1, 12.4.2 12.5.1,

			12.5.2

Unit 3 Learning Outcome (ULO)	CO	BLL	PI addressed
Explain the functioning of different components of the computer system.	1	2	1.6.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Design the building blocks of ALU	5	3	3.7.1,12.4.1, 12.4.2 12.5.1, 12.5.2

Unit 4 Learning Outcome (ULO)	CO	BLL	PI addressed
Explain the functioning of different components of the computer system.	1	2	1.6,12.4, 12.5
Write microprograms	4	3	3.7.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Analyze the performance of processors	3	4	2.8.1, 2.8.2,12.4.1, 12.4.2 12.5.1, 12.5.2

Course Content:

UNIT – I		
Hour	**Syllabus to be covered**	**Mode of delivery**
1	**Basic structure of Computers**: Computer types, Functional Units	**Lecture with a quiz/ Assignments**
2	Basic operational concepts, Bus structures	
3	**Machine instructions and programs:** Numbers, Arithmetic operations an characters	
4	Memory locations and addresses, Memory operations	
5	Instructions and instruction sequencing	
6	Addressing modes	
7	Assembly language , assembler directives, Number Notations	
8	Stacks	
9	Subroutines	
10	Encoding of machine instructions	
UNIT - II		
11	**Input/output organization:** Accessing I/O devices	**Lecture with a quiz/ Assignments**
12	Interrupts-Interrupt hardware, Enabling and Disabling Interrupts	
13	Handling Multiple devices controlling device requests	
14	Exceptions, Direct memory access – Bus Arbitrations	
15	Buses- Asynchronous Bus	
16	Synchronous bus	
17	**Interface Circuits**- Parallel port	
18	serial port	
19	Standard I/O Interfaces –Peripheral component interconnect Bus,	
20	SCSI bus ,USB.	

UNIT - III		
21	**The memory system:** Some Basic concepts,	
22	Semiconductor RAM memories, and cost, Read only memories, speed, size,	
23	cache memories	
24	**Arithmetic Unit:** Addition and subtraction of signed numbers	
25	Design of fast adders	
26	Multiplication of positive numbers, Signed operand multiplication	
27	Fast multiplication	
28	Integer Division	
29	Floating point numbers and operations – IEEE standard for Floating point numbers, Arithmetic operations on Floating point numbers	
30	Implementing Floating point operations	
UNIT - IV		
31	**Basic Processing Unit:** Some fundamental concepts, Execution of complete instruction	**Lecture with a quiz/ Assignments**
32	Hardwired Control	
33	Micro programmed control, Microinstructions,	
34	**Pipelining:** basic concepts, role of cache memory	
35	pipeline performance	
36	**Large computer systems**: forms of parallel processing,	
37	array processor,	
38	the structure of general purpose and multiprocessors systems	
39	**Performance:** Processor Clock, Basic performance equation, pipelining	
40	superscalar operations, Clock rate, Instruction set, compiler, performance measurement	

Review Questions:

Review Questions	ULO	BLL	PI addressed
Explain role of MAR and MBR/MDR registers in memory read and write operations with the help of Instruction -- ADD R1, LOCX	1.1	2	1.6.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Explain following addressing modes with examples: i) Indexed addressing ii) Direct addressing	1.1	2	1.6.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Write an assembly program to multiply two numbers without using MULTIPLICATION instruction.	1.2	3	3.7.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Illustrate use of stack to handle functions.	1.1	2	1.6.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Design 4-bit Adder/Subtractor using full adders as building blocks and additional components if necessary.	3.2	3,4	3.7.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Write sequence of control signals for instruction i) ADD R1, R2 ii) MOV LOCA,R5	4.2	3,4	3.7,12.4,12.5
Explain following terms : i) Micro instruction ii) Micro program iii) Control store iv) Hardwired control unit v) Micro programed control unit.	4.1	2	1.6,12.4, 12.5

Evaluation Scheme:

Assessment	Marks	Weightage
CIE-I	20	20
CIE-II	20	20
Assignments/ Quizzes/ Case Study/ Course Project/ Term Paper/Field Work	10	10
SEE	100	50
Total	150	100

Details of Assignment:

Assignment	Marks (10)	CO	PI	CA	PO
Assignment 1: Write an assembly program	10	4	3.7.1,12.4.1, 12.4.2		4,12
Assignment 2: Design circuit	10	5	3.7.1,12.4.1, 12.4.2		5,12
Assignment 3: Write micro program	10	4	3.7,12.4,12.5		4,12
Assignment 4: Study commercial pipelined and multiprocessor systems	10	1	1.6.1,12.4.1, 12.4.2 12.5.1, 12.5.2		1,12

BASAVESHWAR ENGINEERING COLLEGE (AUTONOMOUS), BAGALKOT

MODEL COURSEPLAN

Title of Course	**:**	**Computer Organization**	**Course Code**	**:**	**22UCS303C**
Credits	**:**	**3**	**Contact Hours/ Week**	**:**	**3**
Total Hours	**:**	**40**	**Tutorial Hours**	**:**	**--**
CIE Marks	**:**	**50**	**SEE Marks**	**:**	**50**
Semester	**:**	**III**	**Year**	**:**	**2023-2024**

Prerequisites:

Course Objectives:

	The Course objectives are:
1	To understand the design and interaction concepts of modern computer subsystem
2	To learn the technique for analyzing the performance of computer system.

Course Outcomes:

	At the end of the course the student should be able to:
1	Explain the design and function of different units of computer
2	Perform the various operations on given data
3	Analyse the execution of the program and different organizations of functional units
4	Develop an assembly programs and micro programs for simple machine instructions
5	Design the basic functional units of computer

Course Articulation Matrix: Mapping of Course Outcomes (CO) with Programme Outcomes (PO) and Programme Specific Outcomes (PSO)

		PO 1	PO2	PO3	PO4	PO5	PO6	PO 7	PO8	PO9	PO 10	PO 11	PO 12	PSO1	PSO2	PSO3
No	**Programme Outcomes / Course Outcomes**															
The students will be able to:																
1	Explain the design and function of different units of Computer	1											1	1		
2	Perform the various operations on given data		3										1	1		2
3	Analyse the execution of the program and different organizations of functional units		2	2									1	1		2
4	Develop assembly programs and micro programs for simple machine instructions			3									1	1		2
5	Design the basic functional units of computer			3									1	1		3

Competencies Addressed in the course and Corresponding Performance Indicators

Programme Outcome:Any of 1 to 12 PO's:

PO 1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialisation for the solution of complex engineering problems.	
Competency	**Indicators**
1.6 Demonstrate competence in engineering fundamentals	1.6.1 Apply engineering fundamentals
1.7 Demonstrate competence in specialized engineering knowledge to the program	1.7.1 Apply theory and principles of computer science and engineering to solve an engineering problem
PO2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	
Competency	**Indicators**

2.8 Demonstrate an ability to execute a solution process and analyze results	2.8.1 Applies engineering mathematics to implement the solution. 2.8.2 Analyze and interpret the results using contemporary tools.
PO 3: Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.	
Competency	**Indicators**
3.6 Demonstrate an ability to generate a diverse set of alternative design solutions	3.6.1 Able to explore design alternatives. 3.6.2 Able to produce a variety of potential design solutions suited to meet functional requirements.
3.7 Demonstrate an ability to select optimal design scheme for further development	3.7.1 Able to perform systematic evaluation of the degree to which several design concepts meet the criteria.
3.8 Demonstrate an ability to advance an engineering design to defined end state	3.8.1 Able to refine architecture design into a detailed design within the existing constraints. 3.8.2 Able to implement and integrate the modules. 3.8.3 Able to verify the functionalities and validate the design.
PO 12: Life-long learning: Recognise the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.	
Competency	**Indicators**
12.4 Demonstrate an ability to identify gaps in knowledge and a strategy to close these gaps	12.4.1 Describe the rationale for the requirement for continuing professional development 12.4.2 Identify deficiencies or gaps in knowledge and demonstrate an ability to source information to close this gap
12.5 Demonstrate an ability to identify changing trends in engineering knowledge and practice	12.5.1 Identify historic points of technological advance in engineering that required practitioners to seek education in order to stay current 12.5.2 Recognize the need and be able to clearly explain why it is vitally important to keep current regarding new developments in your field

PO1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

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

PO9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PSO1: Inculcate analytical, logical, programming and managerial abilities.

PSO2: Apply Software Engineering Principles and Practices to provide software solutions

PSO3: Design and Develop computational systems for real-life applications.

Unit Learning Outcomes (ULO):

Unit 1 Learning Outcome (ULO)	CO	BLL	PI addressed
Explain the functioning of different components of the computer system.	1	2	1.6.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Write assembly programs	4	3	3.7.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Analyze the execution of the program	2	4	2.8.1, 2.8.2,12.4.1, 12.4.2 12.5.1, 12.5.2

Unit 2 Learning Outcome (ULO)	CO	BLL	PI addressed
Explain the functioning of different components of the computer system.	1	2	1.6,12.4, 12.5
Write assembly programs	4	3	3.7.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Design basic I/O interface circuits	5	3	3.7.1,12.4.1, 12.4.2 12.5.1,

			12.5.2

Unit 3 Learning Outcome (ULO)	CO	BLL	PI addressed
Explain the functioning of different components of the computer system.	1	2	1.6.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Design the building blocks of ALU	5	3	3.7.1,12.4.1, 12.4.2 12.5.1, 12.5.2

Unit 4 Learning Outcome (ULO)	CO	BLL	PI addressed
Explain the functioning of different components of the computer system.	1	2	1.6,12.4, 12.5
Write microprograms	4	3	3.7.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Analyze the performance of processors	3	4	2.8.1, 2.8.2,12.4.1, 12.4.2 12.5.1, 12.5.2

Course Content:

UNIT – I		
Hour	**Syllabus to be covered**	**Mode of delivery**
1	**Basic structure of Computers**: Computer types, Functional Units	**Lecture with a quiz/ Assignments**
2	Basic operational concepts, Bus structures	
3	**Machine instructions and programs:** Numbers, Arithmetic operations an characters	
4	Memory locations and addresses, Memory operations	
5	Instructions and instruction sequencing	
6	Addressing modes	
7	Assembly language , assembler directives, Number Notations	
8	Stacks	
9	Subroutines	
10	Encoding of machine instructions	
UNIT - II		
11	**Input/output organization:** Accessing I/O devices	**Lecture with a quiz/ Assignments**
12	Interrupts-Interrupt hardware, Enabling and Disabling Interrupts	
13	Handling Multiple devices controlling device requests	
14	Exceptions, Direct memory access – Bus Arbitrations	
15	Buses- Asynchronous Bus	
16	Synchronous bus	
17	**Interface Circuits**- Parallel port	
18	serial port	
19	Standard I/O Interfaces –Peripheral component interconnect Bus,	
20	SCSI bus ,USB.	

UNIT - III		
21	**The memory system:** Some Basic concepts,	
22	Semiconductor RAM memories, and cost, Read only memories, speed, size,	
23	cache memories	
24	**Arithmetic Unit:** Addition and subtraction of signed numbers	
25	Design of fast adders	
26	Multiplication of positive numbers, Signed operand multiplication	
27	Fast multiplication	
28	Integer Division	
29	Floating point numbers and operations – IEEE standard for Floating point numbers, Arithmetic operations on Floating point numbers	
30	Implementing Floating point operations	
UNIT - IV		
31	**Basic Processing Unit:** Some fundamental concepts, Execution of complete instruction	**Lecture with a quiz/ Assignments**
32	Hardwired Control	
33	Micro programmed control, Microinstructions,	
34	**Pipelining:** basic concepts, role of cache memory	
35	pipeline performance	
36	**Large computer systems**: forms of parallel processing,	
37	array processor,	
38	the structure of general purpose and multiprocessors systems	
39	**Performance:** Processor Clock, Basic performance equation, pipelining	
40	superscalar operations, Clock rate, Instruction set, compiler, performance measurement	

Review Questions:

Review Questions	ULO	BLL	PI addressed
Explain role of MAR and MBR/MDR registers in memory read and write operations with the help of Instruction -- ADD R1, LOCX	1.1	2	1.6.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Explain following addressing modes with examples: i) Indexed addressing ii) Direct addressing	1.1	2	1.6.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Write an assembly program to multiply two numbers without using MULTIPLICATION instruction.	1.2	3	3.7.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Illustrate use of stack to handle functions.	1.1	2	1.6.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Design 4-bit Adder/Subtractor using full adders as building blocks and additional components if necessary.	3.2	3,4	3.7.1,12.4.1, 12.4.2 12.5.1, 12.5.2
Write sequence of control signals for instruction i) ADD R1, R2 ii) MOV LOCA,R5	4.2	3,4	3.7,12.4,12.5
Explain following terms : i) Micro instruction ii) Micro program iii) Control store iv) Hardwired control unit v) Micro programed control unit.	4.1	2	1.6,12.4, 12.5

Evaluation Scheme:

Assessment	Marks	Weightage
CIE-I	20	20
CIE-II	20	20
Assignments/ Quizzes/ Case Study/ Course Project/ Term Paper/Field Work	10	10
SEE	100	50
Total	150	100

Details of Assignment:

Assignment	Marks (10)	CO	PI	CA	PO
Assignment 1: Write an assembly program	10	4	3.7.1,12.4.1, 12.4.2		4,12
Assignment 2: Design circuit	10	5	3.7.1,12.4.1, 12.4.2		5,12
Assignment 3: Write micro program	10	4	3.7,12.4,12.5		4,12
Assignment 4: Study commercial pipelined and multiprocessor systems	10	1	1.6.1,12.4.1, 12.4.2 12.5.1, 12.5.2		1,12

BASAVESHWAR ENGINEERING COLLEGE, BAGALKOTE

COURSE PLAN

Title of Course	:	**Data Structures**	**Course Code**	:	**22UCS304C**
Credits	:	**03**	**Contact Hours/Week**	:	**02**
Total Hours	:	**52**	**Tutorial Hours**	:	**02**
CIE Marks	:	**50**	**SEE Marks**	:	**50**
Semester	:	**III**	**Year**	:	**2023-2024**

1. **Prerequisites:** Strong Foundation in C Programming courses is essential to take up this course.

2. Course Objectives:

At the end of the course student will learn / understand / think / experience / appreciate / have insight / proficiency:	
1	*Advanced C features such as pointers and pointers application, dynamic memory allocation functions and recursion.*
2	*Foundations of data structure operations, design and implementation of data structures to organize data efficiently.*
3	*Designs and implementations of various data structures using abstract data types (ADTs) and without using abstract data types.*
4	*Analysis of variety of problems and selecting suitable data structures to develop the solutions.*
5	*Solving given problems using different data structures.*

3. Course Outcomes:

	At the end of the course the student should be able to:
1	*List and define concepts of pointers, dynamic memory allocation, recursion and data structures.*
2	*Explain advanced C features such as pointers, dynamic memory allocation, recursion and methods of implementing different data structures.*
3	*Implement of various data structures using abstract data type and without using ADTs*
4	*Analyze given problems and identify appropriate data structures to develop the solutions.*
5	*Solve problems using data structures to represent, organize and manipulate data for given problems.*

4. **Course Articulation Matrix:** Mapping of Course Outcomes (CO) with Programme Outcomes (PO) and Programme Specific Outcomes (PSO)

		Program Outcomes(POs)												Program Specific Outcomes(PSOs)		
No.	Course Outcomes	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
The students will be able to:																
1	*List and define concepts of pointers, dynamic memory allocation, and recursion and data structures.*		2	2	2	2								3		2
2	*Explain advanced C features such as pointers, dynamic memory allocation, recursion and methods of implementing different data structures.*		2	2	2	2								2		2
3	*Implement of various data structures using abstract data type and without using ADTs*		3	3	3	3								3		3
4	*Analyze given problems and identify appropriate data structures to develop the solutions.*		3	3	3	3								3		3
5	*Solve problems using data structures to represent, organize and manipulate data for given problems*		3	3	3	3								3		3

5. Competencies Addressed in the course and Corresponding Performance Indicators

5.1 Programme Outcome: Any of 1 to 12 PO's:

PO1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO1. **Engineering knowledge**: *Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.*	
Competency	**Performance Indicators**
1.7 Demonstrate competence in specialized engineering knowledge to the program	1.7.1 Apply theory and principles of computer science and engineering to solve an engineering problem

PO2. Problem analysis: *Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.*	
Competency	**Performance Indicators**
2.5 Demonstrate an ability to identify and formulate complex engineering problem	2.5.1 Evaluate problem statements and identifies objectives 2.5.2 Identify processes/modules/algorithms of a computer-based system and parameters to solve a problem 2.5.3 Identify mathematical algorithmic knowledge that applies to a given problem
2.6 Demonstrate an ability to formulate a solution plan and methodology for an engineering problem	2.6.1 Reframe the computer-based system into interconnected subsystems 2.6.2 Identify functionalities and computing resources. 2.6.3 Identify existing solution/methods to solve the problem, including forming justified approximations and assumptions 2.6.4 Compare and contrast alternative solution/methods to select the best methods 2.6.5 Compare and contrast alternative solution processes to select the best process.
2.7 Demonstrate an ability to formulate and interpret a model	2.7.1 Able to apply computer engineering principles to formulate modules of a system with required applicability and performance. 2.7.2 Identify design constraints for required performance criteria.
2.8 Demonstrate an ability to execute a solution process and analyze results	2.8.1 Applies engineering mathematics to implement the solution. 2.8.2 Analyze and interpret the results using contemporary tools. 2.8.3 Identify the limitations of the solution and sources/causes. 2.8.4 Arrive at conclusions with respect to the objectives.
PO3. Design/development of solutions: *Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.*	
Competency	**Performance Indicators**
3.6 Demonstrate an ability to generate a diverse set of alternative design solutions	3.6.1 Able to explore design alternatives. 3.6.2 Able to produce a variety of potential design solutions suited to meet functional requirements. 3.6.3 Identify suitable non-functional requirements for evaluation of alternate design solutions.

6. Unit Learning Outcomes (ULO):

	Unit Learning Outcome (ULO)	CO	BLL	PI addressed
Unit I	1. Able to describe and apply the knowledge of pointers, memory allocation functions, recursion concepts and Stack data structure to solve different applications. Select array or linked list ADT to implement of data structures to solve problems.	**1, 2, 3, 4, 5**	**1, 2, 3, 4, 5**	**1.7.1, 2.5.1, 2.5.2, 2.5.3, 2.6.3, 2.6.4, 2.7.1, 2.8.1, 2.8.3, 3.6.1,**
Unit II	2. Able to describe and apply the representation and organization of data using stack and queue data structures to solve different applications. Select array or linked list ADT to implement of data structures to solve problems.			

Unit III	3. Able to describe and apply the representation and organization of data using general, circular and doubly linked list to solve different applications. Select linked list ADT to implement of data structures to solve problems.			**3.6.2, 3.7.1, 4.4.2**
Unit IV	4. Able to describe and apply the representation and organization of data using trees and graphs to solve different applications. Select linked list ADT to implement of data structures to solve problems.			

7. Course Content:

Hours	Topic to be covered	Mode of Delivery
	UNIT - I	
01	**Revision on C programming Concepts**	Chalk and talk in classroom/Lecture combined with discussions/Lecture with a quiz/ Tutorial/ Assignments/ Demonstration/ /Asynchronous Discussion
01	**Pointer applications**: Arrays and pointers,	
01	Pointer arithmetic and arrays, passing an array to a function, Using pointers to functions.	
01	Memory allocation functions, Array of pointers, pointers to void and pointers to functions	
01	**Recursion:** iterative and recursive definition iterative and recursive solution, designing recursive functions, limitations of recursion.	
01	**Stacks**: Basic stack operations: Push, Pop, Stack top,	
01	**Stack linked list:** Implementation, Data structure, Stack head, Stack data node, Stack algorithms,	
01	Create Stack, Push Stack, Stack top, Empty Stack, Full Stack, Stack count, Destroy Stack	
01	Create Stack, Push Stack, Stack top, Empty Stack, Full Stack, Stack count, Destroy Stack	
01	**C language implementations:** Insert data, Push Stack , Print Stack, Pop character	
01	**Stack ADT:** Data structure, ADT Implementations, Stack structure, Create stack, Push stack, Pop stack, Stack top, Empty stack, Stack count, Destroy stack	
01	**Stack ADT:** Data structure, ADT Implementations, Stack structure, Create stack, Push stack, Pop stack, Stack top, Empty stack, Stack count, Destroy stack	
01	**Stack Implementation using array**	
	UNIT - II	
01	**Stack applications: Reversing data:** Reverse a list, Convert decimal to binary, Infix to postfix transformation, Evaluating postfix expressions	Chalk and talk in classroom/Lecture combined with discussions/Lecture with a quiz/ Tutorial/ Assignments/ Demonstration/ /Asynchronous Discussion
01	**Stack applications: Reversing data:** Reverse a list, Convert decimal to binary, Infix to postfix transformation, Evaluating postfix expressions	
01	**Queues**: **Queue Operations**: Enqueue, Dequeue, Queue front, Queue rear, Queue example,	
01	**Queues**: **Queue Operations**: Enqueue, Dequeue, Queue front, Queue rear, Queue example,	
01	**Queues**: **Queue Operations**: Enqueue, Dequeue, Queue front, Queue rear, Queue example,	
01	**Queue Linked list design**: Data structure, Queue head, Queue data	

	node, Queue algorithms, Create queue, Enqueue, Dequeue, Retrieving queue data, Empty queue, Full queue, Queue count, Destroy queue	
01	**Queue Linked list design**: Data structure, Queue head, Queue data node, Queue algorithms, Create queue, Enqueue, Dequeue, Retrieving queue data, Empty queue, Full queue, Queue count, Destroy queue	
01	**Queue Linked list design**: Data structure, Queue head, Queue data node, Queue algorithms, Create queue, Enqueue, Dequeue, Retrieving queue data, Empty queue, Full queue, Queue count, Destroy queue	
01	**Queue ADT**: Queue structure, Queue ADT algorithms, Queue Implementation using array,	
01	**Queue ADT**: Queue structure, Queue ADT algorithms, Queue Implementation using array,	
01	**Queue ADT**: Queue structure, Queue ADT algorithms, Queue Implementation using array,	
01	**Queue Applications**	
01	**Queue Applications**	
UNIT - III		
01	**General Linear lists: Basic operations**, Insertion, Deletion, Retrieval, Traversal,	Chalk and talk in classroom/Lecture combined with discussions/Lecture with a quiz/ Tutorial/ Assignments/ Demonstration/ /Asynchronous Discussion
01	**General Linear lists: Basic operations**, Insertion, Deletion, Retrieval, Traversal,	
01	**Implementation:** Data structure, Head node, Data node, Algorithms,	
01	**Implementation:** Create list, Insert node, Delete node, List search, Retrieve node,	
01	**Implementation:** Empty list, Full list, List count, Traverse list, Destroy list,	
01	**List ADT**: ADT functions, Create list, Add node, Internal insertion function	
01	**List ADT**: Search list, Internal search function,	
01	**List ADT**: Retrieve node, Empty list Full list, List count	
01	**List ADT**: Traverse, Destroy list,	
01	**Circular linked lists and Doubly linked lists**: Create list, add node,	
01	**Circular linked lists and Doubly linked lists**: delete node	
01	**Circular linked lists and Doubly linked lists**: Create list, add node, delete node, retrieve node, search list.	
01	**Circular linked lists and Doubly linked lists**: retrieve node, search list.	
UNIT - IV		
01	**Non-Linear lists: Trees: Basic tree concepts**: Terminology, User representation	Chalk and talk in classroom/Lecture combined with discussions/Lecture with a quiz/ Tutorial/ Assignments/ Demonstration/ /Asynchronous Discussion
01	**Binary trees**: Properties, Height of binary trees, Balance, Complete and Nearly complete binary trees	
01	**Binary tree traversals:** Depth-first traversals, Breadth-first traversals,	
01	**Expression Trees**: Infix traversal, Postfix traversal, Prefix traversal	
01	**Expression Trees**: **Huffman code, General trees,**	
01	**Binary search trees**: Basic concepts, **BST operations**: Traversals, Searches	
01	**BST operations**: Insertion Find the smallest and largest node,	
01	**BST operations**: BST search, Insertion, Deletion	
01	**Binary search tree ADT**, Data structure, Head and node structure, Algorithms, Create a BST, Insert a BST,	
01	Internal insert function, Delete a BST, Internal delete function, Retrieve a BST, Internal retrieve function,	

01	Traverse a BST, Empty a BST, Full BST, BST count, Destroy a BST, Internal destroy function.	
01	**Graphs: Basic concepts**, **Operations**: Insert vertex, Delete vertex, Add edge, Delete edge, Find vertex,	
01	**Graph storage structures**: Adjacency matrix, Adjacency list.	

8. Review Questions:

Review Questions	ULO	BLL	PI addressed
1. Define abstract data type.	1	1	1.7.1
2. Why data accessibility is not possible in ADT functions?	1	2	1.7.1
3. Consider the following declaration statements: **void* p;** **int number = 7;** **float price = 23.5;** Write C statements to print values of variable **number** and **price** using pointer p;	1	3	2.5.1
4. Write DisplayStack function using only Stack ADT operations to print contents of stack without changing the stack contents.	1	3	4.4.1
5. What are basic operations of stack? Implement stack ADT operations using linked list.	1	2	2.6.1
6. Using only the operations of queue ADT, write an application called **CopyQueue** that copies the contents of one queue to another.	2	3	3.5.1
7. With the given traversals construct binary tree for the following. i) Preorder: A B M H E O C P G J D K L I N F Inorder**:** H M C O E B A G P K L D I N J F ii) Postoder: F E C H G D B A Inorder: F C E A B H D G	4	4	5.6.2
8. What is queue? Implement circular queue ADT operations using array.	2	2	2.5.1
9. Implement binary tree ADT with the following operations: i) Insert new node ii) Breadth First Traversal of tree iii) To find total number of leaves in a tree iv) To search an element	3	3	2.5.1
10. Write a C program to create employee list. Each node in the list should have Employee Name, Department, and Designation, date of joining, date of birth, salary and pointer to next employee. Write a function to insert an employee, delete an employee and display employee list for a given department.	3	3	2.5.1
11. Implement binary tree ADT with the following operations: a. Insert new node b. Breadth First Traversal of tree c. To find total number of leaves in a tree d. To search an element	3	2	2.5.1
12. With the given traversals construct binary tree for the following. a. Preorder: A B M H E O C P G J D K L I N F Inorder**:** H M C O E B A G P K L D I N J F	4	4	2.5.1

b. Postoder: F E C H G D B A Inorder: F C E A B H D G			

9. Evaluation Scheme:

Assessment	Marks	Weightage
CIE-I	**40**	**20**
CIE-II	**40**	**20**
Assignments/ Quizzes/Case Study/ Course Project/ Term Paper/ Field Work	**10**	**10**
SEE	**100**	**50**
Total	**100**	**100**

10. Details of Assignment:

Assignment	Marks (10)	CO	PI	CA	PO
Assignment 1 – UNIT -I	**10**	**1-4**	**1.7.1, 2.5.1, 2.6.3, 2.6.4, 2.7.1, 2.8.1, 2.8.3, 3.6.1,**	**1.7, 2.5, 2.8, 3.6**	**PO -3**
Assignment 2 – UNIT -I	**10**	**1-5**	**1.7.1, 2.5.1, 2.6.3, 2.6.4, 2.7.1, 2.8.1, 3.6.2, 3.7.1, 4.4.2**	**1.7, 2.5, 2.8, 3.6**	**PO- 3 PO- 4**
Assignment 3 – UNIT -I	**10**	**1-5**	**1.7.1, 2.5.1, 2.8.1, 2.8.3, 3.6.1, 3.6.2, 4.4.2**	**1.7, 2.5, 2.8, 3.6, 4.4**	**PO-3 PO-4 PO-5**
Assignment 4: UNIT -I	**10**	**1-5**	**1.7.1, 2.5.1, 2.6.3, 2.6.4, 2.7.1, 2.8.3, 3.6.1, 3.7.1, 4.4.2**	**1.7, 2.5, 2.8, 3.6, 4.4**	**PO-4 PO-5**

BASAVESHWAR ENGINEERING COLLEGE(AUTONOMOUS), BAGALKOT

Department of Computer Science & Engineering

MODEL COURSE PLAN

Title of Course	**:**	**Digital Systems**	**Course Code**	**:**	**22UCS301C**
Credits	**:**	**3**	**Contact Hours/ Week**	**:**	**3**
Total Hours	**:**	**40**	**Tutorial Hours**	**:**	**0**
CIE Marks	**:**	**50**	**SEE Marks**	**:**	**100**
Semester	**:**	**3**	**Year**	**:**	**2023-24**

Prerequisites:

Basic Electronic Circuits, Number Systems, Boolean algebra and Logic Gates.

Course Objectives:

	The Course objectives are:
1	Make use of simplifying techniques in the design of combinational circuits.
2	Illustrate combinational and sequential digital circuits.
3	Demonstrate the use of flip flops.
4	Design and test registers and counters.
5	Implement the combinational and sequential circuits using Verilog/VHDL programming.

Course Outcomes:

	At the end of the course the student should be able to:
1	Demonstrate the understanding of Boolean algebra.
2	Describe the working of Combinational circuits.
3	Apply the Boolean theorems, K-Map, Q-M and VEM methods to simplify Boolean expressions.
4	Describe the working of Sequential circuits.
5	Simulate combinational and sequential circuits using HDL programming.

22UCS301C	Digital Systems	Credits: 03
L:T:P - 3 : 0 : 0		CIE Marks: 50
Total Hours/Week: 03		SEE Marks: 50

UNIT-I	10 Hrs.

Boolean algebra and Combinational Circuits:

Boolean algebra definition, Principle of Duality, Boolean algebra theorems, Boolean formulas and functions, Normal forms. Minterm canonical form, m-notation, Maxterm Canonical form, M-notation. Manipulation of Boolean expressions. Gates and combinational circuits. Incomplete Boolean functions and don't care conditions, Additional Boolean operations and Gates.

UNIT–II	10 Hrs.

Simplifications of Boolean Expressions:

Formulations of simplification problem, Prime Implicants and Irredundant disjunctive expressions, Prime implicates and Irredundant conjunctive expressions, Karnaugh maps, Using Karnaugh maps to obtain minimal expressions for complete Boolean functions, Minimal expressions of incomplete Boolean functions.

The Quine-McCluskey method of generating Prime Implicants and Prime Implicates, Decimal method for obtaining prime Implicants, Variable-Entered Karnaugh maps.

UNIT–III	10 Hrs.

Logic Design with MSI Components and Programmable Logic Devices:

Binary adders and Subtractor, Decimal adders, Comparators, Decoders, Multiplexers. Programmable logic devices (PLDs), Programmable read only memories (PROMs), Programmable logic arrays (PLAs), Programmable array logics (PALs)

UNIT–IV	10 Hrs.

Flip-Flops and Applications:

The Basic Bistable Element: Lathes, Master-Slave flip-flops (Pulse-Triggered flip-flops), Edge triggered flop-flops, Characteristic equations.

Registers: Serial In Serial Out, Serial In Parallel Out Parallel in Parallel Out, Parallel In Serial Out, Circular, Universal Shift Registers.

Counters: Binary Ripple Counter, Synchronous Binary Counters, Mod and Ring counters. Design of Synchronous Counters.

HDL implementations of combinational and sequential circuits.

Reference Books *

1. D. D. Givone, 8th Edition, 2017, "Digital Principles and Design", McGraw Hill.
2. R. D. Sudhakar Samuel, Revised Edition, 2005, "Logic Design - A simplified approach", Sanguine Technical Publications.
3. Malvino, Leach and Saha, 6th Edition, 2007, "Digital Principles and applications", McGraw Hill.
4. McGraw Hill, 2nd Edition, 2002, "Fundamental of digital Logic with Verilog Design", McGraw Hill.

Course Articulation Matrix: Mapping of Course Outcomes (CO) with Programme Outcomes (PO) and Programme Specific Outcomes (PSO)

		PO1	PO2	PO3	PO4	PO5	PO6	PO 7	PO8	PO9	PO1 0	PO1 1	PO1 2	PSO1	PSO2	PSO3
No	**Programme Outcomes / Course Outcomes**															
The students will be able to:																
1	Demonstrate the understanding of Boolean algebra.	3	1	-	-	-	-	-	-	-	-	-	1	1	-	1
2	Describe the working of Combinational circuits.	2	1	-	-	-	-	-	-	-	-	-	1	1	-	1
3	Apply the Boolean theorems, K-Map, Q-M and VEM methods to simplify Boolean expressions.	2	-	-	-	-	-	-	-	-	-	-	1	1	-	1
4	Describe the working of Sequential circuits.	1	1	2	-	-	-	-	-	-	-	-	1	1	-	2
5	Simulate combinational and sequential circuits using HDL programming.	1	1	2	-	-	-	-	-	-	-	-	1	1	-	2

Competencies Addressed in the course and Corresponding Performance Indicators

Programme Outcome: Any of 1 to 12 PO's:

PO3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

Competency	Indicators
1. Identify complex engineering problem	i. Students should be able to identify problems that require Digital Systems. ii. Identify the domains where Digital Systems may be developed
2. Design a system and identify system requirements	iii. Analysis of the problem, represent the knowledge iv. Encode the knowledge v. Augment the knowledge vi. Analyze the complexity of the solution
3. Consideration for public health, safety	vii. Designing systems for healthcare, cyber security and safety viii. Employ knowledge representation, search, inference, and reasoning abilities
4. Cultural, societal and environmental considerations	ix. Applying the DS for societal needs and environmental protection, especially the problems like conservation of water, conservation of forest, healthcare, telemonitoring, ensuring social and ethical responsibilities, like proctored conduction of examinations

PO1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

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

PO9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Unit Learning Outcomes (ULO):

Unit Learning Outcome (ULO)	CO	BLL	PI addressed
Demonstrate the understanding of Boolean algebra.	**1,2**	**1,2,4**	**i, ii, iii, iv, v, and vi**
Describe the working of Combinational and sequential circuits.	**3**	**3,5**	**vii**
Apply the Boolean theorems, K-Map, Q-M and VEM methods to simplify Boolean expressions.	**4**	**3,4**	**viii**
Simulate combinational and sequential circuits using HDL programming.	**5**	**2, 6**	**ix**

Course Content:

Unit-I **Total Hours: 10**

Boolean algebra and combinational circuits.

Hours	Syllabus to be covered

No	
1-2	Boolean algebra- Definition, principle of duality, Boolean algebra theorems.
3-4	Boolean algebra Boolean formulas and functions.
5-6	Normal forms Canonical Formulas- Minterm canonical formula, m-notation. Maxterm canonical formula, M-notation.
7-8	Manipulation of Boolean expressions.
9-10	Gates and combinational networks. Incomplete Boolean functions and don't care conditions.
11-12	Additional Boolean operations and gates.
13	Introduction to HDL.

Unit-II **Total Hours: 13**

Simplification of Boolean expressions.

Hours No	Syllabus to be covered
14-15	Introduction to Karnaugh-maps.
16-19	Use of Karnaugh-maps to minimize Boolean Expressions..
20	Minimal Expressions of Incomplete Boolean Functions.
21-23	The Quine-McCliskey method of generating prime implicants and prime implicates
24	Decimal method for obtaining prime implicants implicates.
25-26	Variable Entered K-map

Unit-III **Total Hours: 13**

Logic Design using MSI Components, Flip-Flops and applications.

Hours No	Syllabus to be covered
27-29	Binary Adders and Substractor.
30	Comparators.
31-33	Decoders, Encoders and Multiplexers.
34	Basic bistable elements.
35-37	Latches and Master Slave Flip-Flops
38	Edge Triggered Flip-Flops, Characteristic equations.
39	HDL implementation of logic circuits.

Unit-IV **Total Hours: 13**

Registers, Counters, Operational Amplifiers and its Applications

Hours No	Syllabus to be covered
40-41	Registers
42-44	Counters, Design of synchronous counters.
45-46	HDL implementation of flip-flop, registers and counters.
47-48	Introduction to operational amplifiers., Block diagram representation of a typical

	Op-Amp, Equivalent Circuits of an Op-Amps, Ideal Voltage Transfer curve.
49-50	Open Loop Op-Amps Configurations.
51-52	Digital to Analog –Analog to Digital conversion using Op-Amps.

Review Questions:

PART - A

Q. No.	Questions	Marks	BLL	CO	PI
i)	What is full adder? Write the expressions for sum and carry of full adder.	(02)	1	2	1.6.1
ii)	Implement the expression $f = \sum m\ (1,2,5,6)$ using a decoder.	(02)	3	2	1.6.1
iii)	Implement 4:1 MUX using 2:1 MUX only.	(02)	3	2	1.6.1
iv)	Write symbol and truth table for gated SR latch.	(02)	2	4	1.6.1
v)	Draw the circuit diagram for 4-bit circular shift register.	(02)	3	4	1.6.1
	PART – B				
	UNIT - III				
2. a)	Draw a block diagram, write truth table for Full Subtractor, further simplify using Karnaugh map then implement it.	(07)	3	2	1.6.1
b)	Draw a block diagram, write truth table for 2 bit magnitude comparator further simplify using Karnaugh map then implement it.	(08)	3	2	1.6.1
3. a)	Implement the function $f\ (a, b, c, d) = \sum m\ (0,1,5,6,7,9,10,15)$ using 1. 8:1 MUX treat a, b and c as select lines. 2. 4:1 MUX treat a & b as select lines.	(07)	3	2	1.6.1
b)	Draw the circuit diagram and write a truth table for SR latch then explain its working using all four possibilities.	(08)	2	4	1.6.1

Evaluation Scheme:

Assessment	Marks	Weightage
CIE-I	20	20
CIE-II	20	20
Assignments/ Quizzes/Case Study/ Course Project/Term Paper/Field Work	10	10
SEE	50	50
Total	**100**	**100**

Details of Assignment:

Assignment	Marks (10)	CO	PI	CA	PO
Assignment 1: Draw the Karnaugh Maps for the following Boolean functions. 1. $F1 = a\,b + \overline{b}\,c + \overline{a}\,\overline{c}$ 2. $F2 = (a + \overline{b})(b + c)$	2	1	i, ii	1	1,2
Assignment 2: Draw the K-Map and list out all the prime Implicants from the following Boolean expression. $F = \sum m (3, 4, 5, 7, 9, 13, 14, 15)$	2	2	Iii, iv	2	3,4
Assignment 3: Find the minimal sum for the following Boolean expression using K-Map. $F = \Pi M (0, 1, 6)$	2	3	v, vi	2	5
Assignment 4: Design a circuit to square a two bits number.	2	4	vii, viii	3	6
Assignment 5: Design a system which accepts four inputs a, b, c and d and outputs a 1whenever the LSB or MSB goes high, assuming double rail logic.	2	5	ix	4	12

BASAVESHWAR ENGINEERING COLLEGE, BAGALKOTE

COURSE PLAN

Title of Course	:	**Data Structures**	**Course Code**	:	**22UCS304C**
Credits	:	**03**	**Contact Hours/Week**	:	**02**
Total Hours	:	**52**	**Tutorial Hours**	:	**02**
CIE Marks	:	**50**	**SEE Marks**	:	**50**
Semester	:	**III**	**Year**	:	**2023-2024**

1. **Prerequisites:** Strong Foundation in C Programming courses is essential to take up this course.

2. Course Objectives:

At the end of the course student will learn / understand / think / experience / appreciate / have insight / proficiency:	
1	*Advanced C features such as pointers and pointers application, dynamic memory allocation functions and recursion.*
2	*Foundations of data structure operations, design and implementation of data structures to organize data efficiently.*
3	*Designs and implementations of various data structures using abstract data types (ADTs) and without using abstract data types.*
4	*Analysis of variety of problems and selecting suitable data structures to develop the solutions.*
5	*Solving given problems using different data structures.*

3. Course Outcomes:

	At the end of the course the student should be able to:
1	*List and define concepts of pointers, dynamic memory allocation, recursion and data structures.*
2	*Explain advanced C features such as pointers, dynamic memory allocation, recursion and methods of implementing different data structures.*
3	*Implement of various data structures using abstract data type and without using ADTs*
4	*Analyze given problems and identify appropriate data structures to develop the solutions.*
5	*Solve problems using data structures to represent, organize and manipulate data for given problems.*

4. **Course Articulation Matrix:** Mapping of Course Outcomes (CO) with Programme Outcomes (PO) and Programme Specific Outcomes (PSO)

		Program Outcomes(POs)												Program Specific Outcomes(PSOs)		
No.	Course Outcomes	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
The students will be able to:																
1	*List and define concepts of pointers, dynamic memory allocation, and recursion and data structures.*		2	2	2	2								3		2
2	*Explain advanced C features such as pointers, dynamic memory allocation, recursion and methods of implementing different data structures.*		2	2	2	2								2		2
3	*Implement of various data structures using abstract data type and without using ADTs*		3	3	3	3								3		3
4	*Analyze given problems and identify appropriate data structures to develop the solutions.*		3	3	3	3								3		3
5	*Solve problems using data structures to represent, organize and manipulate data for given problems*		3	3	3	3								3		3

5. Competencies Addressed in the course and Corresponding Performance Indicators

5.1 Programme Outcome: Any of 1 to 12 PO's:

PO1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO1. **Engineering knowledge**: *Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.*	
Competency	**Performance Indicators**
1.7 Demonstrate competence in specialized engineering knowledge to the program	1.7.1 Apply theory and principles of computer science and engineering to solve an engineering problem

PO2. Problem analysis: *Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.*	
Competency	**Performance Indicators**
2.5 Demonstrate an ability to identify and formulate complex engineering problem	2.5.1 Evaluate problem statements and identifies objectives 2.5.2 Identify processes/modules/algorithms of a computer-based system and parameters to solve a problem 2.5.3 Identify mathematical algorithmic knowledge that applies to a given problem
2.6 Demonstrate an ability to formulate a solution plan and methodology for an engineering problem	2.6.1 Reframe the computer-based system into interconnected subsystems 2.6.2 Identify functionalities and computing resources. 2.6.3 Identify existing solution/methods to solve the problem, including forming justified approximations and assumptions 2.6.4 Compare and contrast alternative solution/methods to select the best methods 2.6.5 Compare and contrast alternative solution processes to select the best process.
2.7 Demonstrate an ability to formulate and interpret a model	2.7.1 Able to apply computer engineering principles to formulate modules of a system with required applicability and performance. 2.7.2 Identify design constraints for required performance criteria.
2.8 Demonstrate an ability to execute a solution process and analyze results	2.8.1 Applies engineering mathematics to implement the solution. 2.8.2 Analyze and interpret the results using contemporary tools. 2.8.3 Identify the limitations of the solution and sources/causes. 2.8.4 Arrive at conclusions with respect to the objectives.
PO3. Design/development of solutions: *Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.*	
Competency	**Performance Indicators**
3.6 Demonstrate an ability to generate a diverse set of alternative design solutions	3.6.1 Able to explore design alternatives. 3.6.2 Able to produce a variety of potential design solutions suited to meet functional requirements. 3.6.3 Identify suitable non-functional requirements for evaluation of alternate design solutions.

6. Unit Learning Outcomes (ULO):

	Unit Learning Outcome (ULO)	CO	BLL	PI addressed
Unit I	1. Able to describe and apply the knowledge of pointers, memory allocation functions, recursion concepts and Stack data structure to solve different applications. Select array or linked list ADT to implement of data structures to solve problems.	**1, 2, 3, 4, 5**	**1, 2, 3, 4, 5**	**1.7.1, 2.5.1, 2.5.2, 2.5.3, 2.6.3, 2.6.4, 2.7.1, 2.8.1, 2.8.3, 3.6.1,**
Unit II	2. Able to describe and apply the representation and organization of data using stack and queue data structures to solve different applications. Select array or linked list ADT to implement of data structures to solve problems.			

Unit III	3. Able to describe and apply the representation and organization of data using general, circular and doubly linked list to solve different applications. Select linked list ADT to implement of data structures to solve problems.			**3.6.2, 3.7.1, 4.4.2**
Unit IV	4. Able to describe and apply the representation and organization of data using trees and graphs to solve different applications. Select linked list ADT to implement of data structures to solve problems.			

7. Course Content:

Hours	Topic to be covered	Mode of Delivery
	UNIT - I	
01	**Revision on C programming Concepts**	Chalk and talk in classroom/Lecture combined with discussions/Lecture with a quiz/ Tutorial/ Assignments/ Demonstration/ /Asynchronous Discussion
01	**Pointer applications**: Arrays and pointers,	
01	Pointer arithmetic and arrays, passing an array to a function, Using pointers to functions.	
01	Memory allocation functions, Array of pointers, pointers to void and pointers to functions	
01	**Recursion:** iterative and recursive definition iterative and recursive solution, designing recursive functions, limitations of recursion.	
01	**Stacks**: Basic stack operations: Push, Pop, Stack top,	
01	**Stack linked list:** Implementation, Data structure, Stack head, Stack data node, Stack algorithms,	
01	Create Stack, Push Stack, Stack top, Empty Stack, Full Stack, Stack count, Destroy Stack	
01	Create Stack, Push Stack, Stack top, Empty Stack, Full Stack, Stack count, Destroy Stack	
01	**C language implementations:** Insert data, Push Stack , Print Stack, Pop character	
01	**Stack ADT:** Data structure, ADT Implementations, Stack structure, Create stack, Push stack, Pop stack, Stack top, Empty stack, Stack count, Destroy stack	
01	**Stack ADT:** Data structure, ADT Implementations, Stack structure, Create stack, Push stack, Pop stack, Stack top, Empty stack, Stack count, Destroy stack	
01	**Stack Implementation using array**	
	UNIT - II	
01	**Stack applications: Reversing data:** Reverse a list, Convert decimal to binary, Infix to postfix transformation, Evaluating postfix expressions	Chalk and talk in classroom/Lecture combined with discussions/Lecture with a quiz/ Tutorial/ Assignments/ Demonstration/ /Asynchronous Discussion
01	**Stack applications: Reversing data:** Reverse a list, Convert decimal to binary, Infix to postfix transformation, Evaluating postfix expressions	
01	**Queues**: **Queue Operations**: Enqueue, Dequeue, Queue front, Queue rear, Queue example,	
01	**Queues**: **Queue Operations**: Enqueue, Dequeue, Queue front, Queue rear, Queue example,	
01	**Queues**: **Queue Operations**: Enqueue, Dequeue, Queue front, Queue rear, Queue example,	
01	**Queue Linked list design**: Data structure, Queue head, Queue data	

	node, Queue algorithms, Create queue, Enqueue, Dequeue, Retrieving queue data, Empty queue, Full queue, Queue count, Destroy queue	
01	**Queue Linked list design**: Data structure, Queue head, Queue data node, Queue algorithms, Create queue, Enqueue, Dequeue, Retrieving queue data, Empty queue, Full queue, Queue count, Destroy queue	
01	**Queue Linked list design**: Data structure, Queue head, Queue data node, Queue algorithms, Create queue, Enqueue, Dequeue, Retrieving queue data, Empty queue, Full queue, Queue count, Destroy queue	
01	**Queue ADT**: Queue structure, Queue ADT algorithms, Queue Implementation using array,	
01	**Queue ADT**: Queue structure, Queue ADT algorithms, Queue Implementation using array,	
01	**Queue ADT**: Queue structure, Queue ADT algorithms, Queue Implementation using array,	
01	**Queue Applications**	
01	**Queue Applications**	
UNIT - III		
01	**General Linear lists: Basic operations**, Insertion, Deletion, Retrieval, Traversal,	Chalk and talk in classroom/Lecture combined with discussions/Lecture with a quiz/ Tutorial/ Assignments/ Demonstration/ /Asynchronous Discussion
01	**General Linear lists: Basic operations**, Insertion, Deletion, Retrieval, Traversal,	
01	**Implementation:** Data structure, Head node, Data node, Algorithms,	
01	**Implementation:** Create list, Insert node, Delete node, List search, Retrieve node,	
01	**Implementation:** Empty list, Full list, List count, Traverse list, Destroy list,	
01	**List ADT**: ADT functions, Create list, Add node, Internal insertion function	
01	**List ADT**: Search list, Internal search function,	
01	**List ADT**: Retrieve node, Empty list Full list, List count	
01	**List ADT**: Traverse, Destroy list,	
01	**Circular linked lists and Doubly linked lists**: Create list, add node,	
01	**Circular linked lists and Doubly linked lists**: delete node	
01	**Circular linked lists and Doubly linked lists**: Create list, add node, delete node, retrieve node, search list.	
01	**Circular linked lists and Doubly linked lists**: retrieve node, search list.	
UNIT - IV		
01	**Non-Linear lists: Trees: Basic tree concepts**: Terminology, User representation	Chalk and talk in classroom/Lecture combined with discussions/Lecture with a quiz/ Tutorial/ Assignments/ Demonstration/ /Asynchronous Discussion
01	**Binary trees**: Properties, Height of binary trees, Balance, Complete and Nearly complete binary trees	
01	**Binary tree traversals:** Depth-first traversals, Breadth-first traversals,	
01	**Expression Trees**: Infix traversal, Postfix traversal, Prefix traversal	
01	**Expression Trees**: **Huffman code, General trees,**	
01	**Binary search trees**: Basic concepts, **BST operations**: Traversals, Searches	
01	**BST operations**: Insertion Find the smallest and largest node,	
01	**BST operations**: BST search, Insertion, Deletion	
01	**Binary search tree ADT**, Data structure, Head and node structure, Algorithms, Create a BST, Insert a BST,	
01	Internal insert function, Delete a BST, Internal delete function, Retrieve a BST, Internal retrieve function,	

01	Traverse a BST, Empty a BST, Full BST, BST count, Destroy a BST, Internal destroy function.	
01	**Graphs: Basic concepts**, **Operations**: Insert vertex, Delete vertex, Add edge, Delete edge, Find vertex,	
01	**Graph storage structures**: Adjacency matrix, Adjacency list.	

8. Review Questions:

Review Questions	ULO	BLL	PI addressed
1. Define abstract data type.	1	1	1.7.1
2. Why data accessibility is not possible in ADT functions?	1	2	1.7.1
3. Consider the following declaration statements: **void* p;** **int number = 7;** **float price = 23.5;** Write C statements to print values of variable **number** and **price** using pointer p;	1	3	2.5.1
4. Write DisplayStack function using only Stack ADT operations to print contents of stack without changing the stack contents.	1	3	4.4.1
5. What are basic operations of stack? Implement stack ADT operations using linked list.	1	2	2.6.1
6. Using only the operations of queue ADT, write an application called **CopyQueue** that copies the contents of one queue to another.	2	3	3.5.1
7. With the given traversals construct binary tree for the following. i) Preorder: A B M H E O C P G J D K L I N F Inorder**:** H M C O E B A G P K L D I N J F ii) Postoder: F E C H G D B A Inorder: F C E A B H D G	4	4	5.6.2
8. What is queue? Implement circular queue ADT operations using array.	2	2	2.5.1
9. Implement binary tree ADT with the following operations: i) Insert new node ii) Breadth First Traversal of tree iii) To find total number of leaves in a tree iv) To search an element	3	3	2.5.1
10. Write a C program to create employee list. Each node in the list should have Employee Name, Department, and Designation, date of joining, date of birth, salary and pointer to next employee. Write a function to insert an employee, delete an employee and display employee list for a given department.	3	3	2.5.1
11. Implement binary tree ADT with the following operations: a. Insert new node b. Breadth First Traversal of tree c. To find total number of leaves in a tree d. To search an element	3	2	2.5.1
12. With the given traversals construct binary tree for the following. a. Preorder: A B M H E O C P G J D K L I N F Inorder**:** H M C O E B A G P K L D I N J F	4	4	2.5.1

b. Postoder: F E C H G D B A Inorder: F C E A B H D G			

9. Evaluation Scheme:

Assessment	Marks	Weightage
CIE-I	**40**	**20**
CIE-II	**40**	**20**
Assignments/ Quizzes/Case Study/ Course Project/ Term Paper/ Field Work	**10**	**10**
SEE	**100**	**50**
Total	**100**	**100**

10. Details of Assignment:

Assignment	Marks (10)	CO	PI	CA	PO
Assignment 1 – UNIT -I	**10**	**1-4**	**1.7.1, 2.5.1, 2.6.3, 2.6.4, 2.7.1, 2.8.1, 2.8.3, 3.6.1,**	**1.7, 2.5, 2.8, 3.6**	**PO -3**
Assignment 2 – UNIT -I	**10**	**1-5**	**1.7.1, 2.5.1, 2.6.3, 2.6.4, 2.7.1, 2.8.1, 3.6.2, 3.7.1, 4.4.2**	**1.7, 2.5, 2.8, 3.6**	**PO- 3 PO- 4**
Assignment 3 – UNIT -I	**10**	**1-5**	**1.7.1, 2.5.1, 2.8.1, 2.8.3, 3.6.1, 3.6.2, 4.4.2**	**1.7, 2.5, 2.8, 3.6, 4.4**	**PO-3 PO-4 PO-5**
Assignment 4: UNIT -I	**10**	**1-5**	**1.7.1, 2.5.1, 2.6.3, 2.6.4, 2.7.1, 2.8.3, 3.6.1, 3.7.1, 4.4.2**	**1.7, 2.5, 2.8, 3.6, 4.4**	**PO-4 PO-5**

BASAVESHWAR ENGINEERING COLLEGE(AUTONOMOUS), BAGALKOT

Department of Computer Science & Engineering

MODEL COURSE PLAN

Title of Course	**:**	**Digital Systems**	**Course Code**	**:**	**22UCS301C**
Credits	**:**	**3**	**Contact Hours/ Week**	**:**	**3**
Total Hours	**:**	**40**	**Tutorial Hours**	**:**	**0**
CIE Marks	**:**	**50**	**SEE Marks**	**:**	**100**
Semester	**:**	**3**	**Year**	**:**	**2023-24**

Prerequisites:

Basic Electronic Circuits, Number Systems, Boolean algebra and Logic Gates.

Course Objectives:

	The Course objectives are:
1	Make use of simplifying techniques in the design of combinational circuits.
2	Illustrate combinational and sequential digital circuits.
3	Demonstrate the use of flip flops.
4	Design and test registers and counters.
5	Implement the combinational and sequential circuits using Verilog/VHDL programming.

Course Outcomes:

	At the end of the course the student should be able to:
1	Demonstrate the understanding of Boolean algebra.
2	Describe the working of Combinational circuits.
3	Apply the Boolean theorems, K-Map, Q-M and VEM methods to simplify Boolean expressions.
4	Describe the working of Sequential circuits.
5	Simulate combinational and sequential circuits using HDL programming.

22UCS301C	Digital Systems	Credits: 03
L:T:P - 3 : 0 : 0		CIE Marks: 50
Total Hours/Week: 03		SEE Marks: 50

UNIT-I	10 Hrs.

Boolean algebra and Combinational Circuits:

Boolean algebra definition, Principle of Duality, Boolean algebra theorems, Boolean formulas and functions, Normal forms. Minterm canonical form, m-notation, Maxterm Canonical form, M-notation. Manipulation of Boolean expressions. Gates and combinational circuits. Incomplete Boolean functions and don't care conditions, Additional Boolean operations and Gates.

UNIT–II	10 Hrs.

Simplifications of Boolean Expressions:

Formulations of simplification problem, Prime Implicants and Irredundant disjunctive expressions, Prime implicates and Irredundant conjunctive expressions, Karnaugh maps, Using Karnaugh maps to obtain minimal expressions for complete Boolean functions, Minimal expressions of incomplete Boolean functions.

The Quine-McCluskey method of generating Prime Implicants and Prime Implicates, Decimal method for obtaining prime Implicants, Variable-Entered Karnaugh maps.

UNIT–III	10 Hrs.

Logic Design with MSI Components and Programmable Logic Devices:

Binary adders and Subtractor, Decimal adders, Comparators, Decoders, Multiplexers. Programmable logic devices (PLDs), Programmable read only memories (PROMs), Programmable logic arrays (PLAs), Programmable array logics (PALs)

UNIT–IV	10 Hrs.

Flip-Flops and Applications:

The Basic Bistable Element: Lathes, Master-Slave flip-flops (Pulse-Triggered flip-flops), Edge triggered flop-flops, Characteristic equations.

Registers: Serial In Serial Out, Serial In Parallel Out Parallel in Parallel Out, Parallel In Serial Out, Circular, Universal Shift Registers.

Counters: Binary Ripple Counter, Synchronous Binary Counters, Mod and Ring counters. Design of Synchronous Counters.

HDL implementations of combinational and sequential circuits.

Reference Books *

1. D. D. Givone, 8th Edition, 2017, "Digital Principles and Design", McGraw Hill.
2. R. D. Sudhakar Samuel, Revised Edition, 2005, "Logic Design - A simplified approach", Sanguine Technical Publications.
3. Malvino, Leach and Saha, 6th Edition, 2007, "Digital Principles and applications", McGraw Hill.
4. McGraw Hill, 2nd Edition, 2002, "Fundamental of digital Logic with Verilog Design", McGraw Hill.

Course Articulation Matrix: Mapping of Course Outcomes (CO) with Programme Outcomes (PO) and Programme Specific Outcomes (PSO)

		PO1	PO2	PO3	PO4	PO5	PO6	PO 7	PO8	PO9	PO1 0	PO1 1	PO1 2	PSO1	PSO2	PSO3
No	**Programme Outcomes / Course Outcomes**															
The students will be able to:																
1	Demonstrate the understanding of Boolean algebra.	3	1	-	-	-	-	-	-	-	-	-	1	1	-	1
2	Describe the working of Combinational circuits.	2	1	-	-	-	-	-	-	-	-	-	1	1	-	1
3	Apply the Boolean theorems, K-Map, Q-M and VEM methods to simplify Boolean expressions.	2	-	-	-	-	-	-	-	-	-	-	1	1	-	1
4	Describe the working of Sequential circuits.	1	1	2	-	-	-	-	-	-	-	-	1	1	-	2
5	Simulate combinational and sequential circuits using HDL programming.	1	1	2	-	-	-	-	-	-	-	-	1	1	-	2

Competencies Addressed in the course and Corresponding Performance Indicators

Programme Outcome: Any of 1 to 12 PO's:

PO3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

Competency	Indicators
1. Identify complex engineering problem	i. Students should be able to identify problems that require Digital Systems. ii. Identify the domains where Digital Systems may be developed
2. Design a system and identify system requirements	iii. Analysis of the problem, represent the knowledge iv. Encode the knowledge v. Augment the knowledge vi. Analyze the complexity of the solution
3. Consideration for public health, safety	vii. Designing systems for healthcare, cyber security and safety viii. Employ knowledge representation, search, inference, and reasoning abilities
4. Cultural, societal and environmental considerations	ix. Applying the DS for societal needs and environmental protection, especially the problems like conservation of water, conservation of forest, healthcare, telemonitoring, ensuring social and ethical responsibilities, like proctored conduction of examinations

PO1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

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

PO9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Unit Learning Outcomes (ULO):

Unit Learning Outcome (ULO)	CO	BLL	PI addressed
Demonstrate the understanding of Boolean algebra.	**1,2**	**1,2,4**	**i, ii, iii, iv, v, and vi**
Describe the working of Combinational and sequential circuits.	**3**	**3,5**	**vii**
Apply the Boolean theorems, K-Map, Q-M and VEM methods to simplify Boolean expressions.	**4**	**3,4**	**viii**
Simulate combinational and sequential circuits using HDL programming.	**5**	**2, 6**	**ix**

Course Content:

Unit-I **Total Hours: 10**

Boolean algebra and combinational circuits.

Hours	Syllabus to be covered

No	
1-2	Boolean algebra- Definition, principle of duality, Boolean algebra theorems.
3-4	Boolean algebra Boolean formulas and functions.
5-6	Normal forms Canonical Formulas- Minterm canonical formula, m-notation. Maxterm canonical formula, M-notation.
7-8	Manipulation of Boolean expressions.
9-10	Gates and combinational networks. Incomplete Boolean functions and don't care conditions.
11-12	Additional Boolean operations and gates.
13	Introduction to HDL.

Unit-II **Total Hours: 13**

Simplification of Boolean expressions.

Hours No	Syllabus to be covered
14-15	Introduction to Karnaugh-maps.
16-19	Use of Karnaugh-maps to minimize Boolean Expressions..
20	Minimal Expressions of Incomplete Boolean Functions.
21-23	The Quine-McCliskey method of generating prime implicants and prime implicates
24	Decimal method for obtaining prime implicants implicates.
25-26	Variable Entered K-map

Unit-III **Total Hours: 13**

Logic Design using MSI Components, Flip-Flops and applications.

Hours No	Syllabus to be covered
27-29	Binary Adders and Substractor.
30	Comparators.
31-33	Decoders, Encoders and Multiplexers.
34	Basic bistable elements.
35-37	Latches and Master Slave Flip-Flops
38	Edge Triggered Flip-Flops, Characteristic equations.
39	HDL implementation of logic circuits.

Unit-IV **Total Hours: 13**

Registers, Counters, Operational Amplifiers and its Applications

Hours No	Syllabus to be covered
40-41	Registers
42-44	Counters, Design of synchronous counters.
45-46	HDL implementation of flip-flop, registers and counters.
47-48	Introduction to operational amplifiers., Block diagram representation of a typical

	Op-Amp, Equivalent Circuits of an Op-Amps, Ideal Voltage Transfer curve.
49-50	Open Loop Op-Amps Configurations.
51-52	Digital to Analog –Analog to Digital conversion using Op-Amps.

Review Questions:

PART - A

Q. No.	Questions	Marks	BLL	CO	PI
i)	What is full adder? Write the expressions for sum and carry of full adder.	(02)	1	2	1.6.1
ii)	Implement the expression $f = \sum m\ (1,2,5,6)$ using a decoder.	(02)	3	2	1.6.1
iii)	Implement 4:1 MUX using 2:1 MUX only.	(02)	3	2	1.6.1
iv)	Write symbol and truth table for gated SR latch.	(02)	2	4	1.6.1
v)	Draw the circuit diagram for 4-bit circular shift register.	(02)	3	4	1.6.1
	PART – B				
	UNIT - III				
2. a)	Draw a block diagram, write truth table for Full Subtractor, further simplify using Karnaugh map then implement it.	(07)	3	2	1.6.1
b)	Draw a block diagram, write truth table for 2 bit magnitude comparator further simplify using Karnaugh map then implement it.	(08)	3	2	1.6.1
3. a)	Implement the function $f\ (a, b, c, d) = \sum m\ (0,1,5,6,7,9,10,15)$ using 1. 8:1 MUX treat a, b and c as select lines. 2. 4:1 MUX treat a & b as select lines.	(07)	3	2	1.6.1
b)	Draw the circuit diagram and write a truth table for SR latch then explain its working using all four possibilities.	(08)	2	4	1.6.1

Evaluation Scheme:

Assessment	Marks	Weightage
CIE-I	20	20
CIE-II	20	20
Assignments/ Quizzes/Case Study/ Course Project/Term Paper/Field Work	10	10
SEE	50	50
Total	**100**	**100**

Details of Assignment:

Assignment	Marks (10)	CO	PI	CA	PO
Assignment 1: Draw the Karnaugh Maps for the following Boolean functions. 1. $F1 = a\,b + \overline{b}\,c + \overline{a}\,\overline{c}$ 2. $F2 = (a + \overline{b})(b + c)$	2	1	i, ii	1	1,2
Assignment 2: Draw the K-Map and list out all the prime Implicants from the following Boolean expression. $F = \sum m (3, 4, 5, 7, 9, 13, 14, 15)$	2	2	Iii, iv	2	3,4
Assignment 3: Find the minimal sum for the following Boolean expression using K-Map. $F = \Pi M (0, 1, 6)$	2	3	v, vi	2	5
Assignment 4: Design a circuit to square a two bits number.	2	4	vii, viii	3	6
Assignment 5: Design a system which accepts four inputs a, b, c and d and outputs a 1whenever the LSB or MSB goes high, assuming double rail logic.	2	5	ix	4	12