

# **B. Tech. Computer Science and Engineering (Cloud Computing) in collaboration with Google Cloud**

## **Course Structure**

**(2024- 25 admitted batch)**

### **VISION**

*The Apollo University aspires to create global leaders of the future, preparing them for a constantly evolving world and empowering them to build a healthier and more equitable society.*



### **MISSION**

*The Mission is to achieve this vision through transformative education, developed at the intersection of healthcare, science, technology, and management, having roots in innovation & research, in an inclusive culture that fosters practical oriented knowledge, stimulates critical thinking and sustains our legacy of pioneering and excellence, for the benefit of mankind.*

**THE APOLLO UNIVERSITY**

**MURUKAMBATTU**

**CHITTOOR (Dt) 517127**

**ANDHRA PRADESH**



## I - Semester

<b>3 Week Induction Programme</b>						
<b>Course Code</b>	<b>Course Name</b>	<b>Periods per week</b>			<b>Credits</b>	<b>Hours per week</b>
		<b>L</b>	<b>T</b>	<b>P</b>		
BTGT1701	Engineering Physics	3	1	0	4	4
BTGT1702	Engineering Mathematics	3	1	0	4	4
BTGT1801	Problem Solving and Programming with C	3	1	0	4	4
BTGT1101	Communicative English (University Core)	3	0	0	3	3
BTGT1201	University Elective I	3	0	0	3	3
BTGL1701	Engineering Physics Lab	0	0	3	1.5	3
BTGL1801	Problem Solving and Programming with C Lab	0	0	3	1.5	3
--	IT Work shop	0	0	0	0	1
--	Design Thinking	0	0	0	0	1
--	Soft Skills	0	0	0	0	1
--	Mentoring	0	0	0	0	1
--	Technical Seminar	0	0	0	0	1
--	Library	0	0	0	0	1
--	Physical Activity	0	0	0	0	2
--	Extra-curricular activities	0	0	0	0	2
--	Co-curricular activity	0	0	0	0	1
--	Self-Learning	0	0	0	0	1
<b>TOTAL</b>		<b>15</b>	<b>3</b>	<b>6</b>	<b>21</b>	<b>36</b>

## II - Semester

Course Code	Course Name	Periods per week			Credits	Hours per week
		L	T	P		
BTGT1703	Probability & Statistics	3	1	0	4	4
BTGT1802	Basic Electrical and Electronics Engineering	3	1	0	4	4
BTGT1301	Data Structures	3	1	0	4	4
BTGT1302	Python Programming	3	1	0	4	4
BTGT1102	Environmental Studies (University Core)	3	0	0	0	3
BTGT1202	University Elective II	3	0	0	3	3
BTGL1802	Basic Electrical and Electronics Engineering Lab	0	0	2	1	2
BTGL1301	Data Structures Lab	0	0	2	1	2
BTGL1302	Python Programming Lab	0	0	2	1	2
--	Mentoring	0	0	0	0	1
--	Co-curricular activity	0	0	0	0	1
--	Self-Learning	0	0	0	0	1
--	Physical Activity	0	0	0	0	2
--	Extra-curricular activities	0	0	0	0	2
--	Library	0	0	0	0	1
<b>TOTAL</b>		<b>18</b>	<b>4</b>	<b>6</b>	<b>22</b>	<b>36</b>

**III - Semester**

Course Code	Course Name	Periods per week			Credits	Hours per week
		L	T	P		
BTGT2701	Discrete Mathematics and Graph Theory	3	1	0	4	4
BTGT2301	Design and Analysis of Algorithms	3	1	0	4	4
BTGT2302	Object Oriented Programming through Java	3	0	0	3	3
BTGT2801	Digital Logic Design	3	0	0	3	3
	University Core - III	3	0	0	3	3
	University Elective -III	3	0	0	3	3
BTGT2303	Constitution of India	3	0	0	0	3
BTGL2301	Java Programming Lab	0	0	2	1	2
BTGL2801	Digital Logic Design Lab	0	0	2	1	2
--	Mentoring	0	0	0	0	1
--	Co-curricular activity	0	0	0	0	1
--	Self-Learning	0	0	0	0	1
--	Physical Activity	0	0	0	0	2
--	Extra-curricular activities	0	0	0	0	2
--	Soft Skills Training	0	0	0	0	1
--	Certification course	0	0	0	0	1
<b>TOTAL</b>		<b>21</b>	<b>2</b>	<b>4</b>	<b>22</b>	<b>36</b>

### IV - Semester

Course Code	Course Name	Periods per week			Credits	Hours per week
		L	T	P		
BTGT2702	Cloud Computing Foundations	3	1	0	4	4
BTGT2901	Management for Engineers	3	0	0	3	3
BTGT2501	Computer Networks	3	0	0	3	3
BTGT2304	Database Management Systems	3	1	0	4	4
BTGT2305	Operating Systems	3	1	0	4	4
BTGT2307	Computer Organization and Architecture	3	1	0	4	4
BTGT2306	Universal Human Values	3	0	0	0	3
BTGL2302	Database Management Systems Lab	0	0	2	1	2
BTGL2501	Cloud Computing Foundations Lab	0	0	2	1	2
--	Mentoring	0	0	0	0	1
--	Aptitude and Logical Reasoning	0	0	0	0	1
--	Library	0	0	0	0	1
--	Physical Activity	0	0	0	0	2
--	Extra-curricular activities	0	0	0	0	2
--	<b>Internship – I will be Evaluated in V Sem</b>	-	-	-	-	-
<b>TOTAL</b>		<b>21</b>	<b>4</b>	<b>4</b>	<b>24</b>	<b>36</b>

Value Added Courses	
Course Code	Course Name
BTGA0001	Google Cloud Computing Foundation
BTGA0002	Acquiring Data
BTGA0003	Cloud Computing Foundation Course by Microsoft
BTGA0004	Software Programmer - Python

### V - Semester

Course Code	Course Name	Periods per week			Credits	Hours per week
		L	T	P		
BTGT3501	Automata and Compiler Design	3	0	0	3	3
BTGT3301	Full Stack Web Development	3	0	0	3	3
BTGT3502	Cloud Data Analytics	3	0	0	3	3
BTGT3401	Faculty Elective-I	3	0	0	3	3
BTGT3601	Program Elective-I	3	0	0	3	3
BTGM3501	MOOC - I	0	0	0	3	3
BTGT3302	Entrepreneurship and Start up Management	3	0	0	0	0
BTGL3301	Full Stack Web Development Lab	0	0	2	1	2
BTGL3501	Cloud Data Analytics Lab	0	0	2	1	2
BTGI3501	Internship – I Evaluation	0	0	4	2	4
<b>TOTAL</b>		<b>18</b>	<b>0</b>	<b>8</b>	<b>22</b>	<b>26</b>

Faculty Elective -I		Program Elective -I	
Course Code	Course Name	Course Code	Course Name
BTGT3401a	Object Oriented Analysis and Design	BTGT3601a	Software Engineering
BTGT3401b	Software Engineering	BTGT3601b	Cloud Architecture and Services
BTGT3401c	Artificial Intelligence	BTGT3601c	Data Analytics using python
BTGT3401d	Computer Organisation and Architecture	BTGT3601d	Data Warehousing and Mining
BTGT3401e	Linux Programming	BTGT3601e	Artificial Intelligence
BTGT3401f	Advanced Data Structures		
BTGT3401g	Data Warehousing and Mining		
BTGT3401h	Machine Learning		
BTGT3401i	Differential Equations and Vector Calculus		
BTGT3401j	Numerical Methods for Engineers		
BTGT3401k	Mathematical Foundations of Cyber Security		
BTGT3401l	Information Retrieval Systems		
BTGT3401m	Cryptography and Network security		
BTGT3401n	Principles of Programming Languages		
BTGT3401o	Unix and Shell Programming		

BTGT3401p	Introduction to Psychology
BTGT3401q	Industry 4.0
BTGT3401r	Computer Vision

Value Added Courses	
Course Code	Course Name
BTGA0005	Database Engineer Learning Path
BTGA0006	Big Data Foundation Course
BTGA0007	Programming Basics for Blockchain Engineers
BTGA0008	Oracle Cloud Security Administrator

### VI - Semester

Course Code	Course Name	Periods per week			Credits	Hours per week
		L	T	P		
BTGT3303	Bigdata Analytics	3	1	0	4	4
BTGT3503	Cloud management and DevOps	3	0	0	3	3
BTGT3402	Faculty Elective -II	3	0	0	3	3
BTGT3602	Program Elective – II	3	0	0	3	3
BTGT3603	Program Elective – III	3	0	0	3	3
BTGM3502	MOOC-II	0	0	0	3	0
BTGL3502	Bigdata Analytics Lab	0	0	2	1	2
BTGL3503	DevOps Lab	0	0	2	1	2
--	<b>Internship – II will be Evaluated in VII Sem</b>	-	-	-	-	-
<b>TOTAL</b>		<b>15</b>	<b>1</b>	<b>4</b>	<b>21</b>	<b>20</b>

Faculty Elective -II		Program Elective -II	
Course Code	Course Name	Course Code	Course Name
BTGT3402a	Design Patters	BTGT3602a	Machine Learning
BTGT3402b	Cyber Security	BTGT3602b	Intrusion Detection System
BTGT3402c	Software Project Management	BTGT3602c	Cloud Security
BTGT3402d	Agile Software Development	BTGT3602d	High Performance Computing
BTGT3402e	Cloud Computing Tools and Techniques	BTGT3602e	Database Administration
BTGT3402f	Mobile Computing	<b>Program Elective -III</b>	
BTGT3402g	Image Processing	BTGT3603a	Software Project Management

BTGT3402h	Deep Learning	BTGT3603b	Mobile App Development
BTGT3402i	E-Commerce	BTGT3603c	MERN Technologies
BTGT3402j	Block Chain Technology	BTGT3603d	Cyber Security
BTGT3402k	Mathematical Foundations of Data Science	BTGT3603e	Software Requirements Management
BTGT3402l	Transforms and Boundary Value Problems		
BTGT3402m	Optimization Techniques		
BTGT3402n	Information Security		
BTGT3402o	Ethical Hacking		
BTGT3402p	Internet of Things		
BTGT3402q	Embedded Systems		
BTGT3402r	Green Computing		
BTGT3402s	Cloud Security		
BTGT3402t	DevNet		
BTGT3402u	Advanced Computer Networks		
BTGT3402v	Network Security		
BTGT3402w	Fault Tolerant Systems		
BTGT3402x	Computational Intelligence		
BTGT3402y	Data Analytics with Tableau		
BTGT3402z	Human Computer Interaction		

Value Added Courses	
Course Code	Course Name
BTGA0009	Network Engineer Learning Path
BTGA0010	BDA Foundation
BTGA0011	Blockchain Essentials
BTGA0012	Oracle Cloud Operations Engineer

### VII - Semester

Course Code	Course Name	Periods per week			Credits	Hours per week
		L	T	P		
BTGT4501	Multi Cloud and Generative AI	3	0	0	3	3
BTGT4502	Internet of Things	3	0	0	3	3
BTGT4601	Program Elective – IV	3	0	0	3	3
BTGT4602	Program Elective – V	3	0	0	3	3
BTGM4501	MOOC-III	0	0	0	3	0
BTGP4301	Mini Project	0	0	2	1	2
BTGL4501	Generative AI Lab	0	0	2	1	2
BTGL4502	Internet of Things Lab	0	0	2	1	2
BTGI4501	Internship-II Evaluation	0	0	4	2	4
<b>TOTAL</b>		<b>12</b>	<b>0</b>	<b>10</b>	<b>20</b>	<b>22</b>

Program Elective -IV		Program Elective -V	
Course Code	Course Name	Course Code	Course Name
BTGT4601a	Cloud Computing and Virtualization	BTGT4602a	Cloud Data Management
BTGT4601b	Block Chain Technology	BTGT4602b	AWS Cloud Service
BTGT4601c	Natural language Processing	BTGT4602c	Hands on DevOps
BTGT4601d	No SQL Database	BTGT4602d	Game Programming
BTGT4601e	Augmented & Virtual Reality	BTGT4602e	Deep Learning

Value Added Courses	
Course Code	Course Name
BTGA0013	Cloud Developer Learning Path
BTGA0014	Big Data Technology
BTGA0015	Certified Blockchain Developer

**VIII - Semester**

<b>Course Code</b>	<b>Course Name</b>	<b>Periods per week</b>			<b>Credits</b>	<b>Hours per week</b>
		<b>L</b>	<b>T</b>	<b>P</b>		
BTGP4502	Project Work	0	0	24	12	24
BTGV4501	Comprehensive Viva-Voce	0	0	2	1	2
<b>TOTAL</b>		<b>0</b>	<b>0</b>	<b>26</b>	<b>13</b>	<b>26</b>

# **I SEMESTER**

**BTGT1701:**

**ENGINEERING PHYSICS**

**L T P C**

**3 1 0 4**

**Prerequisite: Nil**

**Course Description:**

Engineering Physics is a course made for those keen on research as well as engineering at the same time. It clubs together with the fields of engineering as well as applied sciences and creates a course that helps one gain collective knowledge of the new concepts in the field of physics while also studying and researching their applications at the same time. Engineering Physics course allows you to pursue a wide range of topics at the same time, thus giving you several different options for a career. Being extremely vast and technical, the Engineering Physics course is a much-respected degree that stands apart from other engineering degrees and can be extremely rewarding at the same time.

**Course Objectives:**

1. Introducing fundamental concepts of physics that helps students understand intricacies of physical world around us.
2. To familiarize the students with time dependent quantum systems and their applications
3. To provide the knowledge about lasers to the students. Students should be familiar with the applications of laser in various areas like defence, communication, medical etc
4. Introducing various applications of quantum mechanics such as fibre optics.
5. Semiconductors devices provide an insight to the knowledge of essential electronic devices like LEDs, photodiodes and solar cells.

**UNIT I : Concepts of Quantum Mechanics**

**12 Hrs.**

Introduction-Matter Waves-Particle properties of waves- Wave properties of particles-de Broglie hypothesis-Heisenberg's uncertainty principle - Schrödinger's one-dimensional time independent wave equation- Significance of wave function- Particle in a 1-dimensional potential box- Eigen values of energy-Fermi - Dirac distribution- Effect of temperature on the Fermi Dirac distribution function - Kronig Penny Model-Classification of solids into conductors' semiconductors and insulators.

**UNIT II : Optics**

**12 Hrs.**

**Interference:** Introduction-Superposition principle-constructive interference-Destructive interference-conditions to obtain interference- Interference in thin film by reflection (quantitative analysis)- Conditions to get bright and dark fringes-Newton's rings (qualitative

analysis)- Determination of radius of curvature of the plano convex lens- Determination of refractive index of the liquid-Determination of wavelength of the light source-Applications.

**Diffraction:** Introduction- Fresnel's and Fraunhofer Diffraction- Fraunhofer Diffraction at single slit-Conditions to get principal maximum and minimum intensity positions-Diffraction Grating-Applications

### **UNIT III: Lasers**

**10Hrs.**

Introduction-Characteristics of LASER-Spontaneous and Stimulated Emission of Radiation-Einstein's Coefficients – Population-Population Inversion- mechanisms to obtain population inversion- Meta stable State – components of laser- Ruby Laser- Helium-Neon Laser-Semiconductor Diode Laser-Applications of Lasers.

### **UNIT IV: Fiber Optics**

**12Hrs.**

Introduction-optical fibre- Reflection-Refraction-Total internal reflection-Critical angle-structure and construction of optical fibre-Glass fiber-Plastic fiber-Fiber dimensions-Acceptance Angle and Numerical Aperture-Types of Optical Fibers-Step index optical fibre - Graded index optical fiber- Fiber optic communication system-Advantages of optical fibers in communication- Applications of Optical Fibers.

### **UNIT V: Semiconductors and Semiconductor devices**

**14Hrs.**

Introduction- Semiconductors- Intrinsic and Extrinsic Semiconductors-. Carrier concentration of Intrinsic Semiconductor-Fermi level in intrinsic and extrinsic semiconductor-Carrier concentration of Extrinsic Semiconductor (qualitative)- Drift and Diffusion- Einstein relation-Hall effect and applications- Direct band gap and indirect band gap semiconductors-Semiconductor devices - Formation of a P-N Junction and working of a P-N Junction diode, Energy band Diagram of open circuited P-N Junction Diode, I-V Characteristics of P-N Junction diode. Applications: Light emitting Diode (LED)-Solar Cell - Photo diode.

**Total: 60 Hours**

### **Course Outcomes:**

1. Understanding preliminary concepts of quantum nature and how it give birth to modern engineering.
2. Analyze the intensity variation of light due to interference and diffraction.
3. To gain the knowledge about the fundamentals, principles of laser and knowledge of the various kinds of lasers

4. Develop basic concepts of fiber optics and its applications
5. Apply the knowledge of semiconductors to illustrate the functioning of semiconductor devices

**Text Books:**

1. Engineering Physics, K. Thyagarajan, Mc Graw Hill Education India Pvt.Ltd, 2015.
2. A Textbook of Engineering Physics, M N Avadhanulu, P G Kshirsagar & TVS Arun Murthy, 11<sup>th</sup> edition, S Chand publications Pvt. Ltd, 2018.

**Reference Books:**

1. Introduction to Quantum Mechanics, David J. Griffiths, 3<sup>rd</sup> edition, Pearson Education Cambridge University Press, 2018.
2. Optics, Ajoy Ghatak, 7th edition, McGraw Hill Education India, 2020
3. Concept of Modern Physics, Arthur Beiser, Shobhit Mahajan, S Rai, 7<sup>th</sup> Edition, Tata McGraw Hill, 2017.
4. "Introduction to Solid State Physics" Charles Kittel, 8<sup>th</sup> Edition, Wiley, 2012.

**BTGT1702:**

**ENGINEERING MATHEMATICS**

**L T P C**

**3 1 0 4**

**Course Description:**

This course is about the basic mathematics that is fundamental and essential component in all streams of undergraduate studies in sciences and engineering. The course consists of topics in matrices, quadratic forms, mean value theorems, multivariable calculus and multiple integrals with applications to various engineering problems.

**Course Objectives:**

The course should enable the students to:

1. Instruct the concept of Matrices in solving linear algebraic equations, Eigen values and Eigen vectors.
2. Familiarize the techniques in quadratic forms.
3. The maxima and minima of functions of single and several variables.
4. Illuminate the students in the concept of finding areas and volumes.
5. Equip the students with standard concepts and tools at an intermediate to advanced level mathematics to develop the confidence and ability among the students to handle various real-world problems and their applications.

**UNIT I**

**12 Hrs**

**Matrices:** Rank of a matrix by echelon form, system of homogeneous and non-homogeneous linear equations. Eigen values, Eigen vectors of Matrices, Cayley-Hamilton theorem (without proof), finding inverse and power of a matrix by Cayley-Hamilton theorem.

**UNIT II**

**12 Hrs**

**Quadratic forms:** Symmetric matrix, Orthogonal matrices, Diagonalisation of a matrix by orthogonal process. Quadratic forms and nature of the quadratic forms, reduction of quadratic form to canonical forms by orthogonal transformation.

**UNIT-III**

**12 Hrs**

**Mean value theorems:** Rolle's Theorem, Lagrange's mean value theorem, Cauchy's mean value theorem, Taylor's and Maclaurin theorems with remainders (without proof); Properties and Problems.

**UNIT IV**

**12 Hrs**

**Multivariable calculus:** Partial derivatives, total derivatives, chain rule, change of variables, Jacobians, maxima and minima of functions of two variables, method of Lagrange multipliers.

**UNIT V**

**12 Hrs**

**Multiple integrals:** Double integrals, change of order of integration, double integration in polar coordinates, areas enclosed by plane curves. Evaluation of triple integrals, change of variables between Cartesian and spherical polar co-ordinates.

**Course Outcomes:**

After completion of the course, students will be able to

1. Analyse the solutions of the system of linear equations and find the Eigen values and Eigen vectors of a matrix, which are used to analyse the long-term behaviour of any system.
2. Reduce the quadratic form to canonical form using orthogonal transformations.
3. Utilize mean value theorems to real life problems.
4. Familiarize with functions of several variables, which is useful in optimization.
5. Evaluate multiple integrals in various coordinate systems.

**Text Books:**

1. E. Kreyszig, "Advanced engineering mathematics", John Wiley & Son's publishers, New edition.
2. B. S. Grewal, "Higher engineering mathematics", Khanna publishers, New edition.

**Reference Books:**

1. R. K. Jain and S. R. K. Iyengar, "Advanced Engineering Mathematics, 3/e, Alpha Science International Ltd., 2002.
2. George B. Thomas, Maurice D. Weir and Joel Hass, "Thomas Calculus, 13/e, Pearson Publishers, 2013.
3. Glyn James, "Advanced Modern Engineering Mathematics, 4/e, Pearson publishers, 2012.



## **UNIT V**

**12 Hrs**

**Structures and Union:** Defining structures and Array of Structures, Nested Structures, enum, typedef, Union.

**File handling:** File Handling Functions, File Modes, File Operations.

### **Course Outcomes:**

After completion of the course, the student will be able to:

1. Formulate simple algorithms for arithmetic, logical problems and to translate the algorithms to programs (in C language)
2. Test and execute the programs and correct syntax and logical errors, to implement conditional branching, iteration.
3. Decompose a problem into functions and synthesize a complete program using divide and conquer approach.
4. Use arrays, pointers to formulate algorithms and programs for real-world problems.
5. Apply programming to solve the problems of storing different data's and various operations on files.

### **Text Books:**

1. Programming in C and Data Structures, 2/e, E.Balaguruswamy, Tata McGraw Hill, New Delhi, 2012.
2. E. Balaguruswamy, Programming in ANSI C, 8<sup>th</sup> Edition, Tata McGraw-Hill, 2019.
3. Computer Science, a structured programming approach using C, B.A. Forouzan and R.F. Gilberg, Third edition, Thomson, 2007

### **Reference Books:**

1. R.G. Dromey, —How to Solve it by Computer, Low Price Edition, Pearson Education India, 2008.
2. The C Programming Language, B.W. Kernighan, Dennis M. Ritchie, PHI/Pearson Education, 2015.
3. Let us C by Yashwanth P. Kanetkar 8<sup>th</sup> edition BPB publication, 2016

**BTGT1101:**

**COMMUNICATIVE ENGLISH**

**L T P C**

**3 0 0 3**

**Course Description:**

The creation of the Course is to facilitate Stakeholders in productively using the Language to functional advantage to form meaningful engagements in a social context and influence their professional dynamic.

**Course Objectives:**

1. To heighten awareness of the correct usage of English grammar in Writing and Speaking.
2. To improve their speaking ability in English both in terms of fluency and comprehensibility.
3. To give Oral Presentations and receive Feedback on their performance and increase their reading speed and comprehension of academic articles.
4. To improve their reading fluency skills through extensive reading and enlarge their vocabulary by maintaining a Vocabulary Journal.
5. To strengthen their ability to write academic papers, essays and summaries using the ‘Mind Mapping,’ dynamic.

**UNIT I**

**9 Hrs**

**Vocabulary and Reading:**

Special Features of English Vocabulary, Reading With Purpose; Understanding What is Read; Drawing a Conclusion Based on Inferences, Deduction, Reading Between the Lines, Context, Connotation, Higher Order Thinking; How to Explain Specific Information with Clarity; Defining and Giving Reasons; Giving Directions; Professional Vocabulary.

**UNIT II**

**9 Hrs**

**Basic Grammar:**

Subject-Verb Agreement; Verb Tenses; Active-Passive Voice; Direct and Indirect Speech; Question Tags; Degrees of Comparison; Articles; Avoiding Jargon

**UNIT III**

**9 Hrs**

**Writing:**

Letter Writing; Report Writing; E-Communication, Drafting and Collating Key Information, Taking Notes from Lectures, Reading Materials, Reporting on Minutes of the Meeting, Precis Writing

**UNIT IV**

**9 Hrs**

**Basics of Communication:**

Role of Communication; Purpose of Communication; Barriers to Communication; Verbal and Non-Verbal Communication, Communication at the Workplace; Human Needs and Communication; ‘Mind Mapping’ Communication; E-Communication

## **UNIT V**

**9 Hrs**

### **Presentations:**

Self-Introduction; Individual Presentation; Group Discussions; Debates

### **Course Outcomes:**

1. To review grammatical structures of English and the use of these forms in specific communicative contexts, which include: class activities, homework assignments, reading of texts and writing and functional real-world facets.
2. To improve their accuracy and fluency in producing and understanding spoken and written English and endorse this proficiency in both personal and professional realms.
3. To attain and enhance competence in the four modes of literacy: Writing, Speaking, Reading and Listening.
4. To develop their ability as critical thinkers.
5. To develop presentation skills.

### **ACADEMIC RESOURCES AND REFERENCES**

1. Scholarly Publications - Journals
2. Popular Sources - Newspapers and Magazines
3. Professional Sources
4. Conference Proceedings
5. Government Documents
6. English Communications Skills for Professionals, by Gregory A. Barnes Wren and Martin Grammar

**BTGL1701:**

**ENGINEERING PHYSICS LAB**

**L T P C**

**0 0 3 1.5**

**Prerequisite: Nil**

**Course Description:**

The aim of this course is to make the students gain practical knowledge to co-relate with the theoretical studies and to develop practical applications of engineering materials and use the principle in the right way to implement the modern technology.

**Course Objectives:**

1. To teach students, the importance of physics through involvement in experiments.
2. This lab helps to have knowledge of the world due to constant interplay between observations and hypothesis, experiment and theory in physics.
3. Students will gain knowledge in various areas of physics so as to have real time applications in all engineering streams.

**List of Experiments:**

**Concepts of Experimental Physics**

1. Error-Analysis and Drawing Graph

**Waves and Oscillations**

2. Compound Pendulum
3. Estimation of the Rigidity modulus of a wire using a Torsional Pendulum.
4. Melde's Experiment
5. Resonance in LCR Circuits

**Optics**

6. Determine the thickness of the wire using Wedge shape method
7. Determination of the radius of curvature of the lens by Newton's ring method
8. Determination of dispersive power of prism.
9. Determination of wavelength of laser Source by using diffraction grating.
10. To determine the numerical aperture and acceptance angle of a given optical fibre

**Study experiments**

11. Energy Gap of a semi-conductor.
12. I-V Characteristics (Resistor, Diode, Solar Cell, LDR)

**Course Outcomes:**

After completing this course, the student must demonstrate the knowledge and ability to:

1. Develop analytical/experimental skills and impart prerequisite hands-on experience for engineering laboratories.
2. Understand the need for precise measurement practices for data recording.
3. Understand the principle, concept, working and applications of relevant technologies and comparison of results with theoretical calculations.
4. Analyze the techniques and skills associated with modern scientific tools such as lasers and fiber optics.
5. Develop basic communication skills through working in groups in performing the laboratory experiments and by interpreting the results.

**BTGL1801: PROBLEM SOLVING AND PROGRAMMING WITH C LAB L T P C**

**0 0 3 1.5**

**Course Description:**

This lab course is designed to improve knowledge, logical skills, enhances the analyzing and problem-solving skills which can be used for developing programs in C.

**Learning Objectives:**

The objective of this course is to make students to:

1. Understand the environment and enable to write algorithm and draw the flowchart.
  2. Create the programs using control statements.
  3. Analyse and modularise the problem into sub programs.
  4. Develop programs using the concepts of arrays and strings.
  5. Solve the memory access related problems and can analyse the file handling mechanisms.
1. A) Write an algorithm, draw a flowchart for:
    - i) Finding the area and circumference of a circle of given radius.
    - ii) Finding the volume of a sphere of given radius.
    - iii) Finding the interest on a given principle for a given period of time at a given rate of interest per year.
  - B) Familiarization with programming environment and write a C program for:
    - i) Finding the area and circumference of a circle of given radius.
    - ii) Finding the volume of a sphere of given radius.
    - iii) Finding the interest on a given principle for a given period of time at a given rate of interest per year.
  2. A) Write a C program to calculate the total and percentage of a student for the given marks.  
B) Write a C program to display all the sizes of datatypes in C.  
C) Write a C program to convert an integer into character.
  3. A) Write a C program to find whether a given number is even or odd.  
B) A salesman gets a commission of 5% on the sales if his sales are below Rs.5000/- and a commission of 8% on the sales that exceeds Rs.5000/- together with Rs.250/-. Write an algorithm or a flowchart and develop C program for computing the commission of the salesman, based on his sales.  
C) Write a C program in the menu driven style to perform the operations +, -, \*, /, % between two given integers.
  4. A) i) Write three C programs to print a multiplication table for a given number using while, do-while, and for loops.  
ii) Write a C program to construct Pascal's triangle.  
B) Write a C program to compute the sum of:
    - i)  $1+x+x^2+x^3+\dots+x^n$ , given x and n.
    - ii)  $1 - x^2/2! + x^4/4! - x^6/6! + x^8/8! - x^{10}/10! + \dots$  to n terms where the n<sup>th</sup> term becomes less than 0.0001.

5. A) Write a program to store the numbers given by the user in an array, and then to find the mean, deviations of the given values from the mean, and variance.
- B) Write a program to store user given numbers in an array and find the locations of minimum and maximum values in the array and swap them and display the resulting array.
- C) Write a C program to implement the operations of matrices – addition, subtraction, Multiplication.
6. A) Write C functions for the following:
  - i) A function that takes an integer n as argument and returns 1 if it is a prime number and 0 otherwise.
  - ii) A function that takes a real number x and a positive integer n as arguments and returns  $x^n$ .
  - iii) A function that takes a positive integer n as an argument and returns the n<sup>th</sup> Fibonacci number.
7. Using recursion write C functions for the following:
  - i) Factorial of a non-negative integer n.
  - ii) Greatest Common Divisor of two integers.
8. A) Write a C program that uses functions to perform the insertion of a substring into a given main string from a given position and delete n characters from a given position in a given string.
- B) i) Write a C program to determine whether a given string is palindrome or not.
- ii) Write a C program to count the lines, words and characters in a given text.
9. A) Write a C program to swap two numbers using pointers.
- B) Define a structure and union for student having members roll no., name, class, section, marks. Create an array of 5 students and find the average marks for every student.
10. A) Write a function which copies the contents of one file to another and display the file Contents
- B) Write a C program to reverse the first n characters in a file.
- C) Write a C program to merge two files into a third file.

### **Course Outcomes:**

After completion of the course, the student will be able to

1. Formulate the algorithms and can draw flowcharts for simple problems
2. Write conditional and iterative programs.
3. Develop the programs using sub functions and recursion.
4. Represent data in arrays, strings and can manipulate them.
5. Create files and perform simple operations like reading and writing on files.

### **Text Books:**

1. B.A. Forouzon and R.F. Gilberg, “COMPUTER SCIENCE: A Structured Programming Approach Using C”, Third edition, CENGAGE Learning, 2016.
2. Richard F. Gilberg & Behrouz A. Forouzan, “Data Structures: A Pseudocode Approach with C”, Second Edition, CENGAGE Learning, 2011.
3. Pradip Dey and Manas Ghosh, Programming in C, Oxford University Press, 2nd Edition 2011.

## **II SEMESTER**

**BTGT1703:**  
**C**

**PROBABILITY & STATISTICS**

**L T P**

**3 1 0 4**

**Course Objectives:**

1. To familiarize the students with the foundations of probability and statistical methods.
2. Summarize the basic concepts of data science and its importance in engineering.
3. To impart probability concepts and statistical methods in various applications engineering.
4. Compute conditional probabilities directly and using Baye's theorem, and check for independence of events.
5. Understand the law of large numbers and the central limit theorem and how these concepts are used to model various random phenomena.

**UNIT I**

**12 Hrs**

**Descriptive Statistics:** Statistics Introduction, Measures of Variability (dispersion), Skewness, Kurtosis, correlation, correlation coefficient, rank correlation, principle of least squares, method of least squares, regression lines, regression coefficients and their properties.

**UNIT II**

**12 Hrs**

**Probability:** Probability, probability axioms, addition law and multiplicative law of probability, conditional probability, Baye's theorem, random variables (discrete and continuous), probability density functions, properties.

**UNIT III**

**12 Hrs**

**Probability distributions:** Discrete distribution - Binomial, Poisson approximation to the binomial distribution and their properties. Continuous distribution: normal distribution and their properties.

**UNIT IV**

**12 Hrs**

**Estimation and Testing of hypothesis, large sample tests:** Estimation-parameters, statistics, sampling distribution, point estimation, Formulation of null hypothesis, alternative hypothesis, the critical and acceptance regions, level of significance, two types of errors and power of the test. Large Sample Tests: Test for single proportion, difference of proportions, test for single mean and difference of means. Confidence interval for parameters in one sample and two sample problems.

**UNIT V**

**12 Hrs**

**Small sample tests:** Student t-distribution (test for single mean, two means and paired t-test), testing of equality of variances (F-test),  $\chi^2$  - test for goodness of fit,  $\chi^2$  - test for independence of attributes.

**Course Outcomes:**

Upon successful completion of this course, the student should be able to

1. Compute descriptive statistics using numerical techniques and explain the characteristics through correlation and regression tools.
2. Apply probability theory to find the chances of happening of events.
3. Understand various probability distributions and calculate their statistical moments.
4. Solve the problems on testing of hypothesis on large samples.
5. Solve the problems on testing of hypothesis on small samples.

**Text Books:**

1. Miller and Friends, Probability and Statistics for Engineers, 7/e, Pearson, 2008.
2. S.C. Gupta and V.K. Kapoor, Fundamentals of Mathematical Statistics, 11/e, Sultan Chand & Sons Publications, 2012.

**Reference Books:**

1. S. Ross, a First Course in Probability, Pearson Education India, 2002.
2. W. Feller, an Introduction to Probability Theory and its Applications, 1/e, Wiley, 1968.
3. Peyton Z. Peebles, Probability, Random Variables & Random Signal Principles - McGraw Hill Education, 4th Edition, 2001.



**References Books:**

1. Theory and problems of Basic electrical Engineering-D.P.Kotahari & I.J.Nagrath  
PH  
I.
2. Electronic Devices and Circuits, Millman & Halkias, TMH publications.
3. Electrical Machinery Dr.P.S. Bimbhra, Khanna Publication First Edition 2021.

**BTGT1301:**

**DATA STRUCTURES**

**L T P C**  
**3 1 0 4**

**Course Description:**

The basic idea to introduce this course is to make students to explore several techniques, algorithms to store, organize data and will include linked lists, stacks, queues, graphs and trees. Study and analysing of the various data structures gives complete awareness to use them in various applications.

**Course Objectives:**

The objective of this course is to make students to:

1. Impart the basic concepts of data structures and expected to know the Abstract Data Type, and difference between linear and non-linear data structures
2. Learn implementing concepts about Stacks, Queues
3. Enable them to write algorithms for operations of various linked lists
4. Choose appropriate searching and sorting technique based on the problem to be solved
5. Understand basic concepts about trees, graphs and its applications

**UNIT I**

**12 Hrs**

**Introduction to Data Structures:** Definition of Data Structures, Abstract Data Type, Classification of Data Structures- Linear and Non-Linear, Applications, Data Structure Operations: insertion, deletion, search, traversal.

**UNIT II**

**12 Hrs**

**Stacks and Queues:** Stack ADT and its operations, Array representation of stacks, Stack applications - Reversing Data, Infix to Postfix Transformation, Postfix expression evaluation. Queue ADT and its Operations, Array representation of Queues, Queue Types: Circular Queues, Priority Queue, Dequeue, Applications of Queues. Dynamic Memory Allocation.

**UNIT III**

**12 Hrs**

**Linked Lists:** Concepts of Linked Lists, Types of Linked Lists, Singly Linked lists: Operations & its implementation, Linked representations of Stacks & Queues, Concatenating two lists, Doubly Linked List: Operations & its implementation, Circularly Linked List, Application of Linked Lists.

**UNIT IV**

**12 Hrs**

**Searching and Sorting:** Searching: Linear Search and Binary Search Techniques. Sorting: objectives and properties of different sorting algorithms: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort.

**UNIT V**

**12 Hrs**

**Trees and Graphs:** Trees: Basic Tree Terminologies, Different types of Trees: Binary Trees, Binary Tree Traversals, Applications of Binary Trees, Binary Search Trees, Spanning Trees. Graphs: Basic Terminologies and Representations - Adjacency Matrix, Adjacency List, Graph Traversals, Applications.

**Course Outcomes:**

After completion of this course student will be able to:

1. Demonstrate the concepts of Abstract data type and can differentiate different between linear and non-linear data structures.
2. Implement Stacks and Queues for a given problem.
3. Create programs for Linked lists using various operations.
4. Select the data structure that efficiently model the information in a problem
5. Describe the concepts of trees, graphs and their applications.

**Text Books:**

1. Richard Gilberg, Behrouz Forouzan, —Data Structures: A Pseudocode Approach with C (Data Structures Series), Second Edition, Cengage Learning, 2004.
2. Gav Pai, —Data Structures and Algorithms – Concepts, Techniques and Applications, Tata McGraw Hill, 2008.

**Reference Books:**

1. A. A. Puntambekar, —Data Structures Using C, First Edition, Technical Publications, 2009.
2. E Balagurusamy, —Data Structures Using C, Tata McGraw-Hill Education, 2013.

**BTGT1302:**

**PYTHON PROGRAMMING**

**L T P C**

**3 1 0 4**

**Course Description:**

This course was introduced to make students to learn Python language and programming constructs like loops, conditionals, data structures, and modules. Familiarizes the various tools to run and write Python scripts, and student can develop code quickly, it dives deeper into essential programming topics which helps the students to solve the complex real time problems.

**Course Objectives:**

The objective of this course is to make students to:

1. Know the basic concepts of python and to run the python scripts
2. Develop python programs with different operators, conditions and loops
3. Use python data structures sequences
4. Define and create python functions, modules, packages and make use of them
5. Learn object-oriented programming concepts

**UNIT I**

**12 Hrs**

**Introduction:** History of Python, Need of Python Programming, Applications Basics of Python Programming Using the REPL(Shell), Running Python Scripts, Variables, Assignment, Keywords, Input-Output, Indentation.

**UNIT II**

**12 Hrs**

**Types, Operators and Expressions:** Data Types – Operators - Arithmetic Operators, Comparison (Relational) Operators, Assignment Operators, Logical Operators, Bitwise Operators, Membership Operators, Identity Operators, Expressions and order of evaluations - Control Flow- if, if-elif-else, for, while, break, continue, pass.

**UNIT III**

**12 Hrs**

**Data Structures:** Lists: Slicing and Methods, Tuples: Slicing and Methods, Sets: Slicing and Methods, Dictionaries: Slicing and Methods - Comprehensions, strings and basic operations of strings.

**UNIT IV**

**12 Hrs**

**Functions:** Defining Functions, Calling Functions, Types of Arguments, Anonymous Functions, Fruitful Functions (Function Returning Values), Scope of the Variables in a Function - Global and Local Variables.

**Modules:** Creating modules, import statements, from. Import statement, name spacing.

**Python packages:** Introduction to PIP, Installing Packages via PIP, Using Python Packages.

**UNIT V**

**12 Hrs**

Object Oriented Programming OOP in Python: Classes, 'self-variable', Methods, Constructor Method, Inheritance, Overriding Methods, Data hiding,

**Error and Exceptions:** Difference between an error and Exception, Handling Exception, try except block, Raising Exceptions, User Defined Exceptions.

**Course Outcomes:**

After completion of the course, the student will be able to:

1. Develop and run simple Python programs for solving problems
2. Demonstrate the python programs using various basic concepts
3. Represent compound data using Python lists, tuples, and dictionaries
4. Install python packages and can create and decompose python programs into functions and modules
5. Demonstrate the object-oriented concepts

**Text Books:**

1. Python Programming: A Modern Approach, Vamsi Kurama, Pearson.
2. Learning Python, Mark Lutz, Orielly.

**Reference Books:**

1. Think Python, Allen Downey, Green Tea Press.
2. Core Python Programming, W.Chun, Pearson.
3. Introduction to Python, Kenneth A. Lambert, Cengage.

**BTGT1102:**  
**C**

**ENVIRONMENTAL STUDIES**

**L T P**

**3 0 0 0**

**Course Objectives:**

1. Creating the awareness about environmental problems among people.
2. Imparting basic knowledge about the environment and its allied problems.
3. Developing an attitude of concern for the environment.

**UNIT I**

**9 Hrs**

**Multidisciplinary nature of environmental studies:** Definition, scope and importance; Need for public awareness; **Natural Resources:** Renewable and non-renewable resources; Forest resources: Water resources: Mineral resources; Food resources: Energy resources: Land resources; Equitable use of resources for sustainable lifestyles; Natural resources and associated problems.

**UNIT II**

**9 Hrs**

**Ecosystems:** Concept of an ecosystem.; Introduction, types, characteristic features, structure and function of the following ecosystem :- Forest ecosystem; Grassland ecosystem; Desert ecosystem ;Aquatic ecosystems (ponds, streams, lakes, rivers, oceans, estuaries) Environment Protection Act,1986 ; Public awareness.; Environment and human health.

**UNIT III**

**9 Hrs**

**Biodiversity and its conservation:** Introduction – Definition- genetic, species and ecosystem diversity.; Biogeographical classification of India; India as a mega-diversity nation; Hot-spots of biodiversity.; Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts.; Endangered and endemic species of India; Conservation of biodiversity: In-situ and Ex-situ conservation of biodiversity.

**UNIT IV**

**9 Hrs**

**Environmental Pollution** Definition; Cause, effects and control measures of Air pollution; Water pollution; Soil pollution; Marine pollution; Noise pollution; Thermal pollution and nuclear hazards - Solid waste Management: Causes, effects and control measures of urban and industrial wastes. - Role of an individual in prevention of pollution - Disaster management: floods, earthquake, cyclone and landslides.

**UNIT V**

**9 Hrs**

**Social Issues and the Environment:** From Unsustainable to Sustainable development; Water conservation- rain water harvesting- watershed management; Resettlement and rehabilitation of people; its problems and concerns.; Environmental ethics: Issues and possible solutions.; Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust - Wasteland reclamation.; Consumerism and waste products.; Population and environment

**Course Outcomes:**

The Environmental Studies facilitates students' understanding of complex environmental issues from a problem-oriented, interdisciplinary perspective. After successful completion of the course the students will

1. Understand key concepts from economic, political, and social analysis as they pertain to the design and evaluation of environmental policies and institutions.
2. Appreciate concepts and methods from ecological and physical sciences and their application in environmental problem solving.
3. Appreciate the ethical, cross-cultural, and historical context of environmental issues and the links between human and natural systems.
4. Reflect critically about their roles and identities as citizens, consumers and environmental.

**Text Books:**

1. A Textbook of Environmental Studies (English, Paperback, Asthana D. K.) S.Chand & co, New Delhi
2. Textbook of Environmental Studies for Undergraduate Courses, Erach BHaruchu, UGC, KINDLE Edition, Amazon.

**Reference books:**

1. Encyclopedia of Environment and Society- set of 5 volumes, Sage Publications

**BTGL1802: BASIC ELECTRICAL & ELECTRONICS ENGINEERING LAB L T P C**  
**0 0 2 1**

**Course Objectives:**

1. Understand the Thevenin's networks, DC Machines Speed Control, Brake test, No load test, simple test on single phase transformer and 3-phase Induction motor.
2. Understand the nature and scope of modern electronics, describe physical models of basic components, design and construct simple electronic circuits to accomplish a specific function, e.g., designing amplifiers and understand their capabilities and limitations and make decisions regarding their best utilization in a specific situation.

**Part A: Electrical experiments**

1. Verification of Thevenin's Theorem.
2. Speed control of DC shunt motor by: a) Armature Voltage Control  
b) Field flux control method.
3. Brake test on DC shunt motor.
4. Swinburn's test on DC shunt machine.
5. OC & SC tests on Single – Phase transformer (Predetermination of efficiency and regulation at given power factors).
6. Brake test on 3-phase induction motor (performance characteristics).

**Part B: (Electronics Laboratory experiments)**

1. Identification of various electronic components and Devices  
Specifications, Testing of R, L, C Components (Colour Codes), Potentiometers, Bread Boards.
2. Identification and Specifications of Active Devices like Diodes, BJTs, JFET etc.  
Study and operation of
3. Digital Multi meters
4. Function Signal Generator
5. Regulated Power Supplies
6. Study and Operation of CRO: Measurement of amplitude and frequency. Time Period measurement
7. PN Junction and Zener diode characteristics A. Forward bias B. Reverse bias.
8. Half wave and Full wave Rectifiers.
9. Transistor CE characteristics (Input and Output)
10. FET characteristics
11. CE Amplifier

**Course Outcomes:**

After completing this course, the student will be able to:

1. Understand the Thevenin's Theorem in circuit analysis, speed control methods of DC

motor with and without loading along with performance, working of single-phase transformer at no load conditions, the performance of three phase induction motor, different.

2. Identify, Specify and test R, L, C Components (Colour Codes), Potentiometers, Switches, Coils, Relays.
3. Identify, Specify and test Active Devices, Diodes, BJTs, Low power JFETs.
4. Explain and demonstrate working of PN Junction and Zener diode.
5. Explain and demonstrate working Half and Full wave Rectifier without filters.
6. Demonstrate working of CE characteristics and its application as an amplifier.

**BTGL1301:**

**DATA STRUCTURES LAB**

**L T P C**

**0 0 2 1**

**Course Description:**

This lab course is intended to make students to implement various techniques, algorithms of data structures which can make them more effective designer, developer.

**Course Objectives:**

The main objective of conducting this lab is to enable the students to:

1. Get practical knowledge of data structures learned in the class room.
  2. Extend programming ability using a structured programming approach.
  3. Build and manipulate linear and non-linear data structures, including stacks, queues, linked lists, trees and graphs.
  4. Choose the appropriate data structure to use in solving the typical computer science problems.
- 
1. Write a C program that implements Stack & Queue operations using arrays
  2. Write a C program that uses Stack operations to perform the following
    - a) Converting Infix expression to Postfix expression.
    - b) Evaluating the Postfix expression.
  3. Write a C program that implements Circular Queue and Dequeue operations using Arrays.
  4. Write a C program that implements Stack & Queue operations using Pointers
  5. Write a C program that uses functions to perform the following operations on singly linked list.
    - i) Creation ii) Insertion iii) Deletion iv) Traversal
  6. Write a C program that uses functions to perform the following operations on doubly linked list.
    - i) Creation ii) Insertion iii) Deletion iv) Traversal
  7. Write a C program that uses functions to perform the following operations on circular linked list.
    - i) Creation ii) Insertion iii) Deletion iv) Traversal
  8. Write C programs that use both recursive and non recursive functions to perform the following searching operations for a Key value in a given list of integers:
    - i) Linear search ii) Binary search
  9. Write a C program that implements the following sorting methods to sort a given list of integers in ascending order
    - i) Bubble sort ii) Selection sort iii) Insertion sort iv) Merge Sort v) Quick sort
  10. A) Write a C program that uses functions to perform the following Binary Tree Traversals
    - a) Inorder b) Preorder c) PostorderB) Write a C program to implement the following graph traversals

a) Depth-First Search b) Breadth- First Search

**Course Outcomes:**

After Completing this lab, the student must be able to:

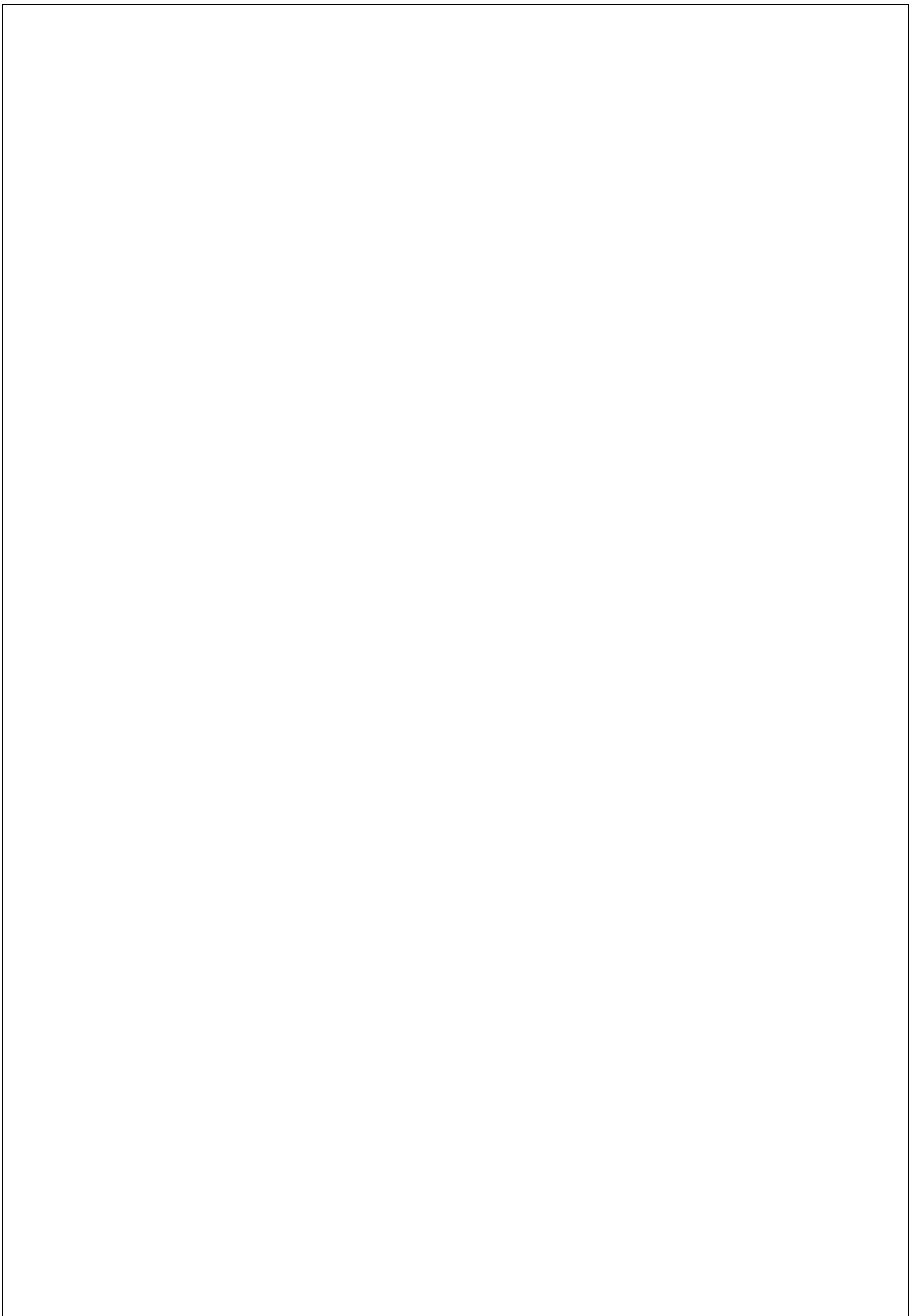
1. Demonstrate the algorithms of various data structures.
2. Emphasize the specification of each data structure as an abstract data type before discussing implementations and application of the structure.
3. Aware of the importance of dynamic allocation and operations of linked lists.
4. Solve the problems by selecting appropriate searching and sorting method.
5. Traverse the binary tree and graph.

**Text Books:**

1. Data Structures: A Pseudo code Approach with C, 2nd Edition, R. F. Gilberg and B. A. Forouzan, Cengage Learning.

**Reference Books:**

1. Fundamentals of Data Structures in C, 2nd Edition, E. Horowitz, S. Sahni and
2. Susan Anderson Freed, Universities Press.
3. Data Structures using C – A. S. Tanenbaum, Y. Langsam, and M. J. Augenstein, PHI/Pearson Education.



**Course Description:**

This lab provides coding exercises on various commonly used concepts like, data structures, writing custom functions, and reading and writing to files. With the gain of the knowledge the students can solve the real-world problems quickly using python constructs and can develop simple html pages.

**Course Objectives:**

The objective of conducting this lab is to enable the students to:

1. Understand python programming basics and paradigm.
2. Learn and develop python looping, control statements and string manipulations.
3. Get familiar with the concepts of data structures and should organize, retrieve the data.
4. Know the concepts of file handling and modular programming.
5. Develop the programs using multi dimension lists and creation of simple html pages.

**1. Basics & operations**

- a) Running instructions in Interactive interpreter and a Python Script.
- b) Write a program to purposefully raise Indentation Error and correct it.
- c) Write a program to compute distance between two points taking input from the user (Pythagorean Theorem).
- d) Write a program add.py that takes 2 numbers as command line arguments and prints its sum.

**2. Control Flow**

- a) Write a Program for checking whether the given number is a even number or not.
- b) Using a for loop, write a program that prints out the decimal equivalents of  $1/2$ ,  $1/3$ ,  $1/4$ ,  $\dots$ ,  $1/10$ .
- c) Write a program using a for loop that loops over a sequence. What is sequence?
- d) Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.

**3. Control Flow - Continued**

- a) Find the sum of all the primes below 200.  
Each new term in the Fibonacci sequence is generated by adding the previous two terms.  
By  
starting with 1 and 2, the first 10 terms will be:  
1, 2, 3, 5, 8, 13, 21, 34, 55, 89,...
- b) By considering the terms in the Fibonacci sequence whose values do not exceed 400, find the sum of the even-valued terms.

**4. DS**

- a) Write a program to count the numbers of characters in the string and store them in a dictionary data structure
- b) Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.
- c) Write a program combining lists that combines these lists into a dictionary.
- d) Write a program to count the frequency of characters in a given file. Can you use character

frequency to tell whether the given file is a Python program file, C program file or a text file?

### **5. Files**

- a) Write a program to print each line of a file in reverse order.
- b) Write a program to compute the number of characters, words and lines in a file.

### **6. Functions**

- a) Write a function ball collide that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding.

Hint: Represent a ball on a plane as a tuple of (x, y, r), r being the radius

If (distance between two balls centers)  $\leq$  (sum of their radii) then (they are colliding)

- b) Find mean, median, mode for the given set of numbers in a list.

### **7. Functions - Continued**

- a) Write a function nearly equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.
- b) Write a function dups to find all duplicates in the list.
- c) Write a function unique to find all the unique elements of a list.

### **8. Functions - Problem Solving**

- a) Write a function cumulative product to compute cumulative product of a list of numbers.
- b) Write a function reverse to reverse a list. Without using the reverse function.
- c) Write a function to compute gcd, lcm of two numbers. Each function shouldn't exceed one line.

### **9. Multi-D Lists**

- a) Write a program that defines a matrix and prints.
- b) Write a program to perform addition of two square matrices.
- c) Write a program to perform multiplication of two square matrices.

### **10 - Modules**

- a) Install packages requests, flask and explore them using (pip).
- b) Write a script that imports requests and fetches content from the page Eg. (Wiki).
- c) Write a simple script that serves a simple HTTP Response and a simple HTML Page.

### **Course Outcomes:**

After completion of the course, the student will be able to:

1. Acquire the knowledge to develop basic Python scripts.
2. Implement the python scripts using conditional and looping statements.
3. Format the data and can organize, retrieve, modify the data easily.
4. Manipulate the files and can simplify the code by modularizing.
5. Install packages and can write scripts to develop simple HTML pages.

### **References:**

1. Think Python, Allen Downey, Green Tea Press.
2. Core Python Programming, W.Chun, Pearson.
3. Introduction to Python, Kenneth A. Lambert, Cengage.

## **III SEMESTER**

**BTGT2701: Discrete Mathematics and Graph Theory****L T P C****3 1 0 4****Course Description:**

The purpose of this course is to understand and use (abstract) discrete structures that are backbones of computer science. Introduce the concepts of mathematical logic and gain knowledge in sets, relations and functions and Solve problems using counting techniques and combinatorics and to introduce generating functions and recurrence relations. Use Graph Theory for solving real world problems.

**Course Objectives:**

1. To explain about the Algebra, Graph theory and Recurrence relations.
2. To demonstrate the application of basic methods of discrete mathematics in Computer Science problem solving.
3. To elucidate solving mathematical problems from algorithmic perspective.
4. To introduce the mathematical concepts which will be useful to study advanced courses Design and Analysis of Algorithms, Theory of Computation, Cryptography and Software Engineering etc.
5. To reveal how solutions of graph theory can be applied to computer science problems

**UNIT-I****12 Hrs****Mathematical Logic:**

Introduction, Statements and Notation, Connectives, Well-formed formulas, Tautology, Duality law, Equivalence, Implication, Normal Forms, functionally complete set of connectives, Inference Theory of Statement Calculus, Predicate Calculus, Inference theory of Predicate Calculus.

**UNIT-II****12 Hrs****Set Theory:**

Basic Concepts of Set Theory, Relations and Ordering, Functions composition of functions, Inverse Functions, Lattices and its properties. Algebraic structures: Algebraic Systems-Examples and General Properties, Semi groups and Monoids, groups, sub groups, homomorphism, Isomorphism.

**UNIT-III****12 Hrs****Elementary Combinatorics:**

Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with Repetitions, Enumerating Permutations with Constrained Repetitions, Binomial Coefficients, The Binomial and Multinomial Theorems, The Principle of Inclusion-Exclusion.

**UNIT-IV****12 Hrs****Recurrence Relations:**

Generating Functions of Sequences, Calculating Coefficients of Generating Functions, Recurrence relations, Solving Recurrence Relations by Substitution and Generating functions, The Method of Characteristic roots, Solutions of Inhomogeneous Recurrence Relations.

**UNIT-V****12 Hrs****Graphs:**

Basic Concepts, Isomorphism and Subgraphs, Trees and their Properties, Spanning Trees, Directed Trees, Binary Trees, Planar Graphs, Euler's Formula, Multigraphs and Euler Circuits, Hamiltonian Graphs, Chromatic Numbers, The Four-Color Problem

**Course Outcomes:**

After completion of the course, students will be able to

1. Apply mathematical logic to solve problems.
2. Understand the concepts and perform the operations related to sets, relations and functions, gain the conceptual background needed and identify structures of algebraic nature.
3. Apply basic counting techniques to solve combinatorial problems.
4. Formulate problems and solve recurrence relations.
5. Apply Graph Theory in solving computer science problems.

**Text Books:**

1. Joe L. Mott. Abraham Kandel and Theodore P. Baker, "Discrete Mathematics for Computer Scientists & Mathematicians", 2<sup>nd</sup> Edition, Pearson, 2008. (For Units III to V).
2. J P Trembly and R Manohar, "Discrete Mathematical Structures with Applications to Computer Science", 1<sup>st</sup> Edition, McGraw Hill, 2017(For Unit I&II).

**Reference Books:**

1. Ralph P. Grimaldi and B.V. Ramana, "Discrete and Combinatorial Mathematics, an Applied Introduction", 5<sup>th</sup> Edition, Pearson, 2016.
2. Narsingh Deo, "Graph Theory with Applications to Engineering", Prentice Hall, 1979.
3. D.S. Malik and M.K. Sen, "Discrete Mathematics theory and Applications", 1<sup>st</sup> Edition, Cenegage Learning, 2012.
4. C L Liu and D P Mohapatra, "Elements of Discrete Mathematics, A computer Oriented approach", 4<sup>th</sup> Edition, MCGRAW-HILL, 2018.

**BTGT2301:**

**Design and Analysis of Algorithms**

**L T P C**

**3 1 0 4**

**Course Description:**

Design and analysis of algorithms is the process of finding the computational complexity of algorithms. It helps to design and analyse the logic on how the algorithm will work before developing the actual code for a program. It focuses on introduction to algorithm, asymptotic complexity, sorting and searching using divide and conquer, greedy method, dynamic programming, backtracking, branch and bound. NP-hard and NP-complete problems. The applications of algorithm design are used for information storage, retrieval, transportation through networks, and presentation to users.

**Learning Objectives:**

1. To analyse the asymptotic performance of algorithms.
2. To apply important algorithmic design paradigms and methods of analysis.
3. To synthesize efficient algorithms in common engineering design situations.
4. Methods and techniques for analysing the correctness and resource requirements of algorithms.

**UNIT I**

**12 Hrs**

**Introduction**

Algorithm - Pseudo Code for Expressing Algorithms - Performance Analysis- Space Complexity - Time Complexity- Asymptotic Notation - Big Oh Notation - Omega Notation - Theta Notation and Little Oh Notation. - Recurrences - Substitution method, Recursion-tree method, Master method.

**UNIT II**

**12 Hrs**

**Disjoint Sets, Divide And Conquer**

Disjoint Sets: Disjoint Set Operations - Union and find Algorithms

Divide and Conquer: General Method - Applications-Binary Search – Quick Sort - Merge Sort- Strassen’s Matrix Multiplication.

**UNIT III**

**12 Hrs**

**Dynamic Programming**

General Method –Applications-Matrix Chain Multiplication - Optimal Binary Search Trees - 0/1 Knapsack Problem - All Pairs Shortest Path Problem - Travelling Sales Person Problem – Reliability Design Problem.

## **UNIT IV**

**12 Hrs**

### **Greedy Method And Backtracking**

Greedy Method: General Method –Applications- Job Sequencing with Deadlines - Knapsack Problem - Minimum Cost Spanning Trees - Single Source Shortest Path Problem - Backtracking:

General Method – Applications-N-Queens Problem - Sum of Subsets Problem - Graph Colouring - Hamiltonian Cycles.

## **UNIT V**

**12 Hrs**

### **Branch And Bound, NP-Hard And NP-Complete Problems**

Branch And Bound: General Method - Applications - Travelling Sales Person Problem - 0/1 Knapsack Problem- LC Branch and Bound Solution - FIFO Branch and Bound Solution.

NP Hard and NP-Complete Problems: Basic Concepts - Non deterministic algorithms - NP – Hard and NP Complete Classes - Cook’s Theorem.

### **Course Outcome:**

1. To gain knowledge of time complexity, space complexity and recurrence methods.
2. To design searching and sorting algorithms along with divide and conquer method and disjoint sets.
3. To apply Dynamic Programming design technique for problem solving.
4. To apply Greedy and back tracing design technique for problem solving
5. To understand the branch and bound algorithms for solving the complex problems

### **Text Book(s)**

1. Ellis Horowitz, Satraj Sahni, Sanguthevar Rajasekharan, “Fundamentals of Computer Algorithms”, Universities Press, 2nd Edition, 2015.
2. Alfred V. Aho, John E. Hopcroft, Jeffrey D, “The Design And Analysis Of Computer Algorithms”, Pearson India, 1st Edition, 2013.

### **Reference Books**

1. Knuth Donald E, “Art of Computer Programming: Fundamental Algorithms Volume 1 - Fundamental Algorithms”, Third Edition, Pearson Publishers, 2011.
2. Levitin A, “Introduction to the Design and Analysis of Algorithms”, Pearson Education, 3rd Edition, 2012.
3. Thomas H. Cormen, C.E. Leiserson, R L.Rivest and C. Stein, Introduction to Algorithms , Third edition, MIT Press, 2009
4. Jon Kleinberg, ÉvaTardos , Algorithm Design, Pearson education, 2014

**BTGT2302: Object Oriented Programming through Java**

**L T P C**  
**3 0 0 3**

**Course Description:**

The course introduces object-oriented programming (OOP) concept and OOP based software development methodology. In this course the students learn object-oriented nature of Java. Java is a class based and OOP language used to demonstrate and implement appropriate concepts and techniques.

**Learning Objectives:**

1. Understand the basic concepts of Object-Oriented Programming principles.
2. Acquire knowledge of object-oriented paradigm in the Java programming language.
3. Develop programs belongs to concepts of Inheritance, Data Abstraction.
4. Understand handling exceptions, streams, applets.
5. Use Java in a variety of technologies and on different platforms.

**UNIT I: An Overview of Java:**

Evolution of java, Object Oriented Programming, Data types, Type conversion and casting, Automatic Type Promotion in Expressions, Arrays, strings, Arithmetic Operators, Operator Precedence

**Control Statements:** Java's selection Statements, Iteration statements, Jump Statements.  
Case study

**UNIT II: Introducing classes:**

Class Fundamentals, Declaring Objects, Introducing Methods, Constructors, Argument Passing, Returning Objects, Recursion, Using Object as Parameter, This Keyword

Overloading Methods and constructors, static and final keywords, Introducing Nested and Inner classes, exploring the String class, Using Command line Arguments  
Case study

**UNIT III: Inheritance:**

Inheritance Understanding Inheritance, Types of Inheritance Significance of Inheritance, Constructor call in Inheritance, use of super keyword in Java, Polymorphism, Understanding Polymorphism, Significance of Polymorphism in Java, Method Overriding, Dynamic Method Dispatching.

**Data Abstraction:**

Basics of Data Abstraction, Understanding Abstract classes, Understanding Interfaces, Multiple Inheritance Using Interfaces, Packages, Access protection, Importing Packages  
Case study

**UNIT-IV: Exception Handling:**

Exception handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch clauses, Nested try statements, throw, throws, finally, Java Built-in Exceptions, creating your own exception subclasses, Chained Exceptions, Three Recently added Exceptions features, Using Exceptions.

**Stream:** I/O basics, Reading Console input, writing console Output, The Printwriter class, Reading and writing files, automatically closing a file. Applet class, lifecycle of an Applet.  
Case study

**UNIT-V:**

**Event Handling** - The Delegation event model- Events, Event sources, Event Listeners, Event classes, Handling mouse and keyboard events, Adapter classes, Inner classes, Anonymous Inner classes.

**Swing (JFC)** - Introduction, difference between and swing, Components Hierarchy, Panes, Individual Swings Components JLabel, JButton, JTextField, JTextArea, JScrollPane, JComboBox

**Networking** - Introduction to JDBC Connectivity, Remote Method Invocation (RMI), Client server communications

Case study

**Course Outcomes:**

1. Understand the basics of java programming like variables, operators, and control statements.
2. Design and implement object oriented (OO) programs to solve problems.
3. Analyze the importance of Inheritance and learn how to implement data abstraction.
4. Understand the concept of Exception Handling and streams.
5. Learn to create user interface using swing components.

**Text Books:**

1. "Java the Complete Reference", Herbert Schildt, MCGRAW HILL Education, 9<sup>th</sup> Edition, 2016.
2. "Understanding Object – Oriented Programming with Java", T. Budd, updated edition, Pearson Education, 2000.

**Reference Books:**

1. "Programming with Java" T.V.Suresh Kumar, B.Eswara Reddy, P.Raghavan Pearson Edition.
2. "Object Oriented Programming through Java", P. Radha Krishna, 1<sup>st</sup> edition, Universities Press, 2007.
3. "Java Programming and Object Oriented Application Development", 1<sup>st</sup> Edition, Cengage Learning, 2006.
4. "An Introduction to programming and OO Design using Java", updated edition, Pearson Education, 2000.

**BTGT2801:**

**DIGITAL LOGIC DESIGN**

**L T P C**

**3 0 0 3**

**Course Description**

Foundation in design and analysis of the operation of digital gates. Design and implementation of combinational and sequential logic circuits. Concepts of Boolean algebra, Karnaugh maps, flip-flops, registers, and counters and VHDL Programming for Digital Circuits.

**Learning objectives:**

1. Provides introduction to logic designs and the basic building blocks used in digital systems.
2. To understand the number systems and codes, Boolean algebra, and logic gates.
3. To minimize the logical functions using Boolean algebra, K-maps, tabular method, and also to understand combinational circuits.
4. To understand different sequential circuits and different Arithmetic circuits.
5. To understand VHDL programming for digital circuits

**UNIT-I**

**8 Hrs**

Number Systems and Codes: Decimal, Binary, Octal, and Hexa-decimal number systems and their conversions, ASCII code, Excess-3 code, Gray code, Complement representation of negative numbers: Signed Magnitude, One's complement method, Two's complement method, Binary Arithmetic.

**UNIT-II**

**8 Hrs**

Logic Simplification and Combinational Logic Design: Review of Boolean Algebra and De Morgan's Theorem, SOP & POS forms, Canonical forms, Karnaugh maps up to 6 variables, Binary codes, Code Conversion.

**UNIT-III**

**10 Hrs**

Combinational Logic Design: Analysis of combinational circuits, Design Procedure – Binary Adder, Subtractor, BCD Adder, multiplier, comparator, decoders, encoders, multiplexers, demultiplexers, Code Converters.

**UNIT-IV**

**10 Hrs**

Sequential Circuits: Latches: RS latch and JK latch, Flip-flops: RS, JK, D, T flip flops, Master-slave flip flops, Edge-triggered flip-flops. Shift registers, Universal Shift register, ripple counters, synchronous counters, Ring counter, Johnson counter, Up-Down counter.

**UNIT-V**

**9 Hrs**

VLSI Design flow: Design entry, Schematic, HDL, different modeling styles in VHDL, Data types and objects, Dataflow, Behavioural and Structural Modeling, Synthesis and Simulation VHDL constructs and codes for combinational and sequential circuits.

**Course Outcomes:**

At the end of the course the student will be able to

1. Understand significance of number systems, conversions, binary codes.

2. Apply different simplification methods for minimizing boolean functions.
3. Illustrate knowledge on design of various combinational circuits.
4. Illustrate the concept of sequential logic design, analyze the operation of flip-flops, registers, and counters.
- 5 Discuss the Use of HDL & appropriate EDA tools for digital logic design and simulation.

**Text Books:**

1. M. Morris Mano and Michael D. Ciletti, Digital Design, 4th Edition, Pearson Education, 2013.
2. D.V. Hall, "Digital Circuits and Systems", Tata McGraw Hill, 1989

**Reference Books:**

1. Anand Kumar, Switching Theory and Logic Design, 2nd Edition, PHI, 2014.
2. Charles Roth, "Digital System Design using VHDL", Tata McGraw Hill 2nd edition 2012.
3. R.P. Jain, "Modern digital Electronics", Tata McGraw Hill, 4th edition, 2009.

**Learning Objectives:**

1. Learn to use object orientation to solve problems and use java language to implement them.
2. To experiment with the syntax and semantics of java language and gain experience with java programming.
3. Illustrate inheritance concepts for reusing the program.
4. Design and develop GUI based applications using applets.
5. Understand and use of GUI components.

**LIST OF EXPERIMENTS:**

- 1) Preparing and practice – Installation of Java software, study of any Integrated development environment, sample programs on operator precedence and associativity, class and package concept, scope concept, control structures, constructors and destructors. Learn to compile, debug and execute java programs.
- 2) a) Write Java program(s) to display n prime numbers.  
b) Write Java program(s) to multiply two matrices.
- 3) a) Write a Java program to create a student class with following fields
  - i. Hall ticket number
  - ii. Student Name
  - iii. Department Create 'n' number of Student objects where 'n' value is passed as input to constructor.b) Write a Java program to demonstrate String comparison using == and equals method.
- 4) Write a program in JAVA to demonstrate the method and constructor overloading.
- 5) a) Demonstrate the implementation of inheritance (multilevel, hierarchical and multiple) by using extend and implement keywords.  
b) Write a java program to implement the concept of dynamic method dispatch.
- 6) a) Write a java program to implement stack concept using interface.  
b) Write a java program to demonstrate the differences between access specifiers.
- 7) a) Write a java program to create a user defined exception that displays an error message when user enters an integer value greater than n.  
b) Write a program to develop an applet that displays a simple message.
- 8) a) Write a java program to split a given text file into n parts. Name each part as the name of the original file followed by .part<n> where n is the sequence number of the

part file.

- b) Write a java program to create a super class called Figure that receives the dimensions of two dimensional objects. It also defines a method called area that computes the area of an object. The program derives two subclasses from Figure.
- 9) a) Design a simple calculator which performs all arithmetic operations.  
b) Write a java program to handle keyboard and mouse events
- 10) a) Understand the process of graphical user interface design and implementation using swings.  
b) Write a Program that creates User Interface to perform Integer Divisions. The user enters two numbers in text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the result field when the divide button clicked. If Num1 or Num2 were not integer, the program would throw a NumberFormatException, If Num2 is Zero, and the program would throw an ArithmeticException. Display the Exception in message box.

#### **Course Outcomes:**

After the completion of this course the student be able to:

1. Solve the problem using object-oriented approach and design solutions which are robust.
2. Develop portable programs which work in all environments using classes and objects.
3. Write the programs illustrating inheritance concept.
4. Create user defined exceptions and user friendly interfaces.
5. Ability to design user interface which can handle keyboard and mouse events.

**LIST OF EXPERIMENTS (Hardware)**

1. Realization of Boolean Expressions using Gates
2. Design and realization of logic gates using universal gates
3. Design and realization of a 4 – bit Gray to Binary and Binary to Gray Converter
4. Verify the functionality of Mux and De-Mux ICs
5. Verify the functionality of Encoder and Decoder ICs
6. Design and realization of 4-bit comparator
7. Verify the functionality of D & T Flip-Flop ICs
8. Verify the functionality of S-R & J-K Flip-Flop ICs
9. Mod-N counter using 7490 and 74190.
10. Shift register IC 7495.

**LIST OF EXPERIMENTS (Hardware)**

## VHDL Programming:

1. Write structural and dataflow VHDL models for
  - a) Half adder / Half subtractor
  - b) Full adder / Full subtractor
2. Write a VHDL program in structural model for
  - a) 16:1 mux realization
  - b) 3:8 decoder realization through 2:4 decoder
3. Write a VHDL program in behavioural model for
  - a) 3:8 decoder
  - b) 8:3 encoder
4. Write a VHDL program in structural and behavioural models for 8 bit asynchronous up-down counter
5. Write a VHDL program in structural model for 8-bit Universal Shift Register.

**Course Outcomes:** At the end of this course students will have the ability to

1. Construct Basic combinational Circuits and Verification of its functionality
2. Construct Sequential Circuits and Verification of its functionality
3. Write structural, behavioural and data flow models for digital circuits
4. Simulate VHDL models of digital circuits

## **IV SEMESTER**

**Course Description:**

This course provides a comprehensive overview of cloud computing technologies and their practical applications. Students will learn about various cloud service models, architecture patterns, storage options, networking configurations, and data processing techniques. The course covers major cloud platforms, such as Google Cloud Platform (GCP), and emphasizes hands-on experience through practical exercises and case studies.

**Course Objectives:**

1. Understand the fundamental concepts of cloud computing.
2. Explore different cloud architectures and compare them with traditional on-premises setups.
3. Gain proficiency in using cloud services such as virtual machines, managed databases, and serverless computing.
4. Develop skills in building scalable and resilient applications using cloud-native technologies like Kubernetes and Cloud Run.
5. Analyse real-world scenarios to identify appropriate cloud solutions and formulate strategies for effective cloud adoption and migration.

**UNIT-I 12 Hrs**

Cloud Computing - Cloud Architecture - Cloud Vs Traditional Architecture - IaaS, PaaS, SaaS - Cloud Shell and console - Google Cloud APIs - Virtual Machine - Event Driven programs with cloud functions - Cloud run - google Kubernetes.

**UNIT-II 12 Hrs**

Storage options in the cloud - Structure and unstructured data - SQL Managed Services - Cloud SQL - Cloud Spanner - No SQL - Cloud End Points - Pub/Sub - Big data managed services.

**UNIT-III 12 Hrs**

Networking in cloud - VMs - VPC - Multiple VPC - HTTP load balancer - Cloud Armor - Cloud Monitoring - Security in cloud - Encryption options- Identity aware proxy.

**UNIT-IV 12 Hrs**

Dataproc - Data flow - ML in cloud - Vertex AI - AutoML - Pre build APIs.

**UNIT-V 12 Hrs**

Case Study on cloud: Analyze case studies of cloud adoption and migration projects in different industries (e.g., healthcare, finance, e-commerce).

**Course Outcomes:**

After completion of the course, students will be able to

1. Demonstrate knowledge of cloud computing principles, including service models, deployment models, and architectural patterns.
2. Evaluate the advantages and disadvantages of cloud computing compared to traditional on-premises environments.
3. Utilize cloud platforms like Google Cloud Platform to deploy, manage, and scale applications effectively.
4. Design and implement cloud-native solutions leveraging containerization, serverless computing, and managed services.
5. Apply cloud computing concepts and techniques to solve practical problems and address business challenges in various industries.

**Text Books:**

1. Google Cloud Computing Foundations, Google Skill Development, [https://www.cloudskillsboost.google/course\\_templates/153](https://www.cloudskillsboost.google/course_templates/153).
2. Thomas Erl, Ricardo Puttini, and Zaigham Mahmood, "Cloud Computing: Concepts, Technology & Architecture", 1st Edition, PHI, 2023.
3. JJ Geewax , "Google Cloud Platform in Action", 1st edition, Manning Publications, 2018.

**Reference Books:**

1. Borko Furht and Armando Escalante, "Handbook of Cloud Computing", Springer-Verlag New York Inc, 2014
2. Srinivasan, "Cloud Computing: A Practical Approach for Learning and Implementation", 1st edition, Pearson Education India, 2014.

**BTGT2901**

**Management for Engineers**

**L T P C**

**3 0 0 3**

**Course Description:** This course is intended to help the students to learn the basic concepts and functions of management and its role in the performance of an organization and to understand various decision-making approaches available for managers to achieve excellence. Learners shall have a broad view of different functional areas of management like operations, human resource, finance and marketing.

**Learning Objectives:**

1. Introducing fundamental concepts of management that helps student understand intricacies of management world around us.
2. Introducing management aspects of nature starting from the basics and critical concepts of different nature,
3. Introducing decision making and functional areas of management
4. Introducing various aspects related to entrepreneurship

**UNIT I**

**12 Hrs**

**Introduction to management Theory**

Introduction to management theory, Management Defined, Characteristic of Management, Management as an art-profession, System approaches to Management, Task and Responsibilities of a professional Manager, Levels of Manager and Skill required.

**UNIT II**

**12Hrs**

**Management and Organization**

Management Process, Planning types, Mission, Goals, Strategy, Programmes, Procedures, Organising, Principles of Organisation, Delegation, Span of Control, Organisation Structures, Directing, Leadership, Motivation, Controlling.

**UNIT III**

**12Hrs**

**Productivity and decision making**

Concept of productivity and its measurement; Competitiveness; Decision making process; decision making under certainty, risk and uncertainty; Decision trees; Models of decision making

**UNIT IV**

**12Hrs**

**Functional areas of management**

Introduction to functional areas of management, Operations management, Human resources management, Marketing management, financial management.

## **UNIT V**

**12Hrs**

### **Entrepreneurship**

**Introduction to entrepreneurship** -Entrepreneurship in India, Role of Government, Business plans, Corporate social responsibility, Patents and Intellectual property rights.

#### **Course Outcomes:**

1. Basic knowledge of management aspects.
2. Understanding of organisations.
3. Understanding of the productivity and decision-making concepts.
4. Understanding entrepreneurship.

#### **Text Books:**

1. H. Koontz, and H. Weihrich, Essentials of Management: An International Perspective. 8th ed., McGraw-Hill, 2009.
2. Robbins, S. P., Judge, T., Vohra, N. (2013). Organizational Behavior. India: Pearson.

#### **Reference Books:**

1. P. Kotler, K. L. Keller, A. Koshy, and M. Jha, Marketing Management: A South Asian Perspective. 14th ed., Pearson, 2012.
2. M. Y. Khan, and P. K. Jain, Financial Management, Tata-McGraw Hill, 2008.
3. R. D. Hisrich, and M. P. Peters, Entrepreneurship: Strategy, Developing, and Managing a New Enterprise, 4th ed., McGraw-Hill Education, 1997.
4. D. J. Sumanth, Productivity Engineering and Management, McGraw-Hill Education, 1985.
5. K. Ashwathappa, 'Human Resources and Personnel Management', TMH, 3 rd. edition, 2005.
6. R. B. Chase, Ravi Shankar and F. R. Jacobs, Operations and Supply Chain Management, 14th ed. McGraw Hill Education (India), 2015.
7. P C Tripathi and P N Reddy, Principles of management, TMH, 4th edition, 2008.

**BTAT3402**

**COMPUTER NETWORKS**

**L T P C**

**3 0 0 3**

**Course Description:**

Computer Networks is a fundamental course that provides an in-depth understanding of the principles, protocols, and technologies that underpin modern computer networks. This course is designed to equip students with the knowledge and skills required to design, implement, and manage computer networks, a critical component of today's information technology landscape.

**Course Objectives:**

1. Identify the components required to build various types of networks, understand the OSI and TCP/IP architectures.
2. Understanding MAC sublayer, Routing Protocols and IP addressing.
3. Learn the use of TCP, UDP protocols.
4. Develop skills to apply the concept of application layer protocols.
5. Understanding WLANs.

**UNIT I Introduction, Physical and Data Link Layer**

**9 Hrs**

Fundamental Components of Networks – Network Topologies - OSI & TCP/IP References models – Transmission Media – Switching – Router - Data link layer design issues - Error detection and corrections – Stop and Wait protocol – Sliding window protocol – Go-back N protocol.

**UNIT II MAC and Network Layer**

**9 Hrs**

MAC - Ethernet - Network layer design issues - Routing algorithms (RIP, OSPF, BGP, DSDV, DSR, AODV) – Congestion control algorithms – IPv4 and IPv6 address.

**UNIT III Transport Layer**

**9 Hrs**

Elements of transport protocols – Congestion Control - The Internet Transport Protocols: UDP, TCP: Introduction – TCP Service model –TCP protocol – TCP segment header – TCP Connection Establishment, Connection release – TCP sliding window – TCP timer management – TCP Congestion control, STCP.

**UNIT IV Application Layer**

**9 Hrs**

Domain Name System- Electronic mail (SMTP, POP3, IMAP, MIME) - WWW – HTTP – FTP - Web Services - Firewalls.

**UNIT V WLANs**

**9 Hrs**

Introduction to Wireless, Components of WLANs, WLAN Operation, Infrastructure Networks, Ad-hoc Networks, WLAN Threats, Secure WLANs, WLAN Configuration, Routing Concepts, IP Static Routing.

**Course Outcomes:**

1. Demonstrate knowledge on fundamentals of network components and topologies, Analyze the OSI and TCP/IP stack and the different protocols in Data Link layer.
2. Demonstrate various types of routing techniques and design the different routing protocols for wired / wireless.
3. Demonstrate uses of datagram delivery.
4. Apply the different strategies Operations of DNS, FTP, HTTP, Email Protocols, SNMP
5. Deploy secure wireless LANs, manage WLAN components, and understand WLAN operation, security, and deployment.

**Text Books:**

1. Andrew S. Tanenbaum, David J. Wetherall, "Computer Networks", Pearson Education, New Jersey, 5th edition, 2011.
2. Behrouz A. Forouzan, "Data Communications and Networking", Tata McGraw Hill, New Delhi, 5th edition, 2012.

**Reference Books:**

1. Computer Networking: Principles, Protocols and Practice, Olivier Bonaventure, CreateSpace Independent Publishing Platform, 1st Edition, 2014
2. Computer Networks: A Systems Approach, Larry L. Peterson and Bruce S. Davie, Morgan Kaufmann, 5th Edition, 2011.
3. Network Security Essentials: Applications and Standards, William Stallings, Pearson, 7th Edition, 2021.
4. Routing and Switching Essentials Companion Guide, Cisco Networking Academy, Cisco Press, 1st Edition, 2013

**BTGT2304**

**Database Management Systems**

**L T P C**

**3 1 0 3**

**Course Description:**

Database Management Systems is intended to provide a clear understanding of fundamentals with emphasis on their applications to create and manage large data sets. It emphasizes on technical overview of database software to retrieve data from database. This includes database design principles, normalization, and concurrent transaction processing, security, recovery and file organization techniques. This will provide adequate knowledge to understand future evolutions of data technologies.

**Course Objectives:**

The objective of this course is to make students to:

1. Discuss the basic database concepts, applications, data models, schemas and instances and design Entity Relationship (E-R) model for a database.
2. Demonstrate the use of integrity constraints, relational algebra operations and relational calculus.
3. Describe the basics of SQL, construct queries using SQL, SQL functions, trigger and cursor concepts in PL/SQL.
4. Understand reasoning about functional dependency and to make the students to identify the role of normalization in database management systems.
5. To present the students with the knowledge of Transaction, concurrency and recovery strategies of DBMS

**UNIT I: Database Systems and Entity Relationship Modeling**

**12 Hrs**

Database System Applications - Purpose of Database Systems - View of Data - Database Languages - Database Users and Administrators - Various Components of overall Database System Structure- Data Models-The Entity-Relationship Model - Attributes and Entity Sets - Relationship Sets - Entity-Relationship Diagrams.

**UNIT II: Relational Data Model**

**12 Hrs**

Introduction to the Relational Model - Integrity Constraints -Relational algebra selection and projection - set operations- renaming – joins -division-examples of algebra queries- Tuple Relational Calculus - Domain Relational Calculus-Expressive power of algebra and calculus.

**UNIT III: Introduction to SQL**

**12 Hrs**

Structured Query Language (SQL): Introduction to SQL -Data types - Data Definition language commands - Data Manipulation language Commands and Data control Language Commands - Candidate Key - Primary key - Foreign key - Select Clause - Where Clause - Logical Connectivity's – AND - OR - Range Search - Pattern Matching - Order By - Group By - Set Operations – Union - Intersect and Minus - Aggregate Functions - Join Operations. PL/SQL: Control Structures - functions - Triggers and Cursors.

#### **UNIT IV: Normalization**

**12 Hrs**

Introduction to Schema Refinement - Properties of Decompositions – Functional Dependencies – Attribute closure - Normal Forms - First - Second - Third – BCNF - Basic definitions of MVDs and JDs - Fourth and Fifth normal forms.

#### **UNIT V: Transaction Processing Concepts and Concurrency Control Techniques 12 Hrs**

Transaction Concept - Transaction States - Implementation of Atomicity and Durability - Serializability - Recoverability - Concurrent Executions - Lock-Based Protocols for Concurrency Control - Time Stamp-Based Protocol for Concurrency Control - Multiple Granularity Recovery System: Recovery and Atomicity – Log based Recovery – Recovery with Concurrent Transactions. **Query Processing, Query Optimization, Storage Management**

#### **Course Outcomes:**

After completion of the course - the student will be able to:

1. Demonstrate knowledge on Data models and Database Languages and Design Entity Relationship model for a database
2. Analyse the relational database theory, and be able to write relational algebra and relational calculus expressions for queries
3. Analyse and evaluate the databases using SQL DML/DDI Commands
4. Analyse databases using normal forms to provide solutions for real time applications
5. Understand the properties of transactions in a database system, Analyse concurrency control techniques for handling concurrent transactions and understand recovery of data from failures

#### **Text Books:**

1. Henry F. Korth, Silberchatz, Sudarshan, “Database System Concepts”, 7/e, 2019, Tata McGraw-Hill, New York.
2. Raghurama Krishnan, “Database Management System”, 2/e, 2000, Tata McGraw Hill, New York.

#### **Reference Books:**

1. Ramez Elmasri, Shamkant B. Navathe, "Fundamental Database Systems", Pearson Education, 6<sup>th</sup> Edition, 2014.
2. Database Principles Fundamentals of Design Implementation and Management, Carlos Coronel, Steven Morris, Peter Robb, Cengage Learning.
3. Database Management Systems, 3/e, Raghurama Krishnan, Johannes Gehrke, TMH
4. “Database Systems, Concepts, Design and Applications” by S.K.Singh, Pearson Education.

**BTGT2305**

**Operating Systems**

**L T P C**

**3 1 0 4**

**Course Description:**

This course examines the important problems in operating system design and implementation. This discussion will cover the trade-offs that can be made between performance and functionality during the design and implementation of an operating system. Particular emphasis will be given to three major OS subsystems: process management memory management (segmentation, paging, swapping), and file systems; and on operating system support for distributed systems.

**Learning Objectives:**

1. To introduce the operating system concepts, designs and provide skills required to implement the services.
2. To describe the trade-offs between conflicting objectives in large scale system design.
3. To develop the knowledge for application of the various design issues and services.

**Unit 1: Introduction to OS**

**10 Hrs**

Functionality of OS - OS design issues - Structuring methods (monolithic, layered, modular, micro-kernel models) - Abstractions, processes.

**OS Principles:** System calls, System/Application Call Interface – Protection: User/Kernel modes - Interrupts

**Unit 2: Scheduling**

**14 Hrs**

Processes Scheduling - CPU Scheduling: Pre-emptive, non-pre-emptive – Multiprocessor scheduling – Deadlocks - Resource allocation and management - Deadlock handling mechanisms: prevention, avoidance, detection, recovery.

**Concurrency:** Inter-process communication, Synchronization - Implementing synchronization primitives (Peterson's solution, Bakery algorithm, synchronization hardware) - Semaphores – Classical synchronization problems, Monitors: Solution to Dining Philosophers problem

**Unit 3: Memory Management**

**12 Hrs**

Main memory management, Memory allocation strategies, Virtual memory: Hardware support for virtual memory (caching, TLB) – Paging - Segmentation - Demand Paging - Page Faults – Page Replacement -Thrashing

**Unit 4 : Storage Management, Protection and Security**

**12 Hrs**

Disk structure and attachment – Disk scheduling algorithms (seek time, rotational latency based)-System threats and security – Policy vs mechanism - Access vs authentication

**Unit 5: System protection**

**12 Hrs**

Access matrix – Capability based systems - OS: performance, scaling, future directions in mobile OS, Recent Trends.

**Course Outcome:**

1. To interpret the evolution of OS functionality, structures and layers. System calls and to find the stages of various process states.
2. To design a model scheduling algorithm to compute various scheduling criteria.
3. To apply and analyse communication between inter process and synchronization techniques. Implement page replacement algorithms, memory management problems and segmentation.
4. To differentiate the file systems for applying different allocation and access techniques.

**Text Book(s)**

1. Abraham Silberschatz, Peter B. Galvin, Greg Gagne-Operating System Concepts, Wiley(2018).
2. Stallings (2006), Operating Systems, Internals and Design Principles, 5th edition, Pearson Education, India.

**Reference Books**

1. Ramez Elmasri, A.Gil Carrick, David Levine, Operating Systems, A Spiral Approach- McGrawHill Higher Education (2010).
2. Remzi H. Arpaci-Dusseau, Andrea C. Arpaci-Dusseau, Operating Systems, Three Easy Pieces, Arpaci-Dusseau Books, Inc (2015).
3. Andrew S. Tanenbaum, Modern Operating Systems, Pearson, 4 th Edition (2016).
4. William Stallings, Operating Systems: Internals and Design Principles, Pearson, 9 th Edition (2018).

**BTGT2307**

**Computer Organization and Architecture**

**L T P C**

**3 1 0 3**

**Course Description:**

Computer Architecture is a core curriculum of computing and engineering programs, which aims to cultivate students' abilities towards the basic architecture of uniprocessor in terms of system performance. Studies of recent high-performance microprocessors, vector processors, and memory systems design to improve cache performance, multi-programming and I/O design improvements like: program I/O VS Interrupt driven I/O. The design optimization of PAL or PLA's. Finally on top of technology the implementation of FPGA and CPLD's

**Learning Objectives:**

- 1.To provide the understanding on basic structure and digital computer operations
- 2.To develop the skills on different communication methods with I/O devices and interfaces
- 3.To deliver the knowledge of the architecture and assembly language programming of 8085 microprocessor.
- 4.To understand the peripherals and their interfacing with 8085 microprocessor.
5. To understand various multiarchitecture.

**UNIT I : Introduction To Computer Systems**

**12 Hrs**

Overview of Organization and Architecture - Functional components of a computer -Registers and register files-Interconnection of components Organization of the von Neumann machine and Harvard Architecture-Performance of processor. Data representation, fixed and floating point and error detecting codes.

**UNIT II: Fundamentals Of Computer Architecture**

**12Hrs**

Introduction to ISA (Instruction Set Architecture)- Instruction formats- Instruction types and addressing modes- Instruction execution (Phases of instruction cycle)- Assembly language programming-Subroutine call and return mechanisms-Single cycle Data path design-Introduction to multi cycle data path-Multi cycle Instruction execution.

**UNIT III: Micro Programmed Control**

**12Hrs**

Control memory, address sequencing, micro program example, and design of control unit. Computer Arithmetic: Fixed point representation of numbers algorithms for arithmetic operations: multiplication (Booths, Modified Booths) - division (restoring and non- restoring) - Floating point representation with IEEE standards and algorithms for common arithmetic operations- Representation of non-numeric data (character codes).

**UNIT IV: The Memory System****12Hrs**

Memory systems hierarchy-Main memory organization-Types of Main memory-memory interleaving and its characteristics and performance- Cache memories: address mapping-line size-replacement and policies- coherence- Virtual memory systems- TLB Reliability of memory systems- error detecting and error correcting systems.

**UNIT V: Device Subsystems****12Hrs**

External- RAID Levels- I/O Performance. Performance Enhancements: Classification of models - Flynn's taxonomy of parallel machine models (SISD, SIMD, MISD, MIMD)- Introduction to Pipelining- Pipelined data path-Introduction to hazards. Contemporary issues: Recent Trends: Multiprocessor architecture: Overview of Shared Memory architecture, Distributed architecture.

**Course Outcomes:**

1. To recognize the functionalities of computer architecture and its components.
2. To apply various basic algorithms and operations to solve complex arithmetic problems complying with IEEE standards.
3. To apply the concepts of memory management for analysis of system performance.
4. To identify the I/O components of computer architecture and their performance.
5. To describe pipelining mechanisms and recognize different parallel machine models.

**Text Books:**

1. M. Morris Mano, Computer System Architecture, 3<sup>rd</sup> edition, PHI, India, 2006.
2. Carl Hamacher, Zvonks Vranesic, SafeaZaky, Computer Organization, 5th edition, McGraw Hill, New Delhi, India, 2010.

**Reference Books:**

1. William Stallings, Computer Organization and Architecture, designing for performance, 8th edition, Prentice Hall, New Jersey, 2010.
2. Andrew S. Tanenbaum, Structured Computer Organization, 5th edition, Pearson Education Inc, New Jersey, 2006. 3. Sivarama P. Dandamudi, Fundamentals of Computer Organization and Design, Springer Int. Edition, USA, 2003
3. David A. Patterson and John L. Hennessy Computer Organization and Design-The Hardware/Software Interface 5th edition, Morgan Kaufmann, 2013
4. Tanenbaum, A.S ,2016. Structured computer organisation., Pearson Education, India

**Course Description:**

This Laboratory course introduces the query language for design and development of a database by using various software's such as SQL, ORACLE etc. It provides practice on built-in SQL functions using languages like DDL, DCL, DML and TCL to create and manage database systems and perform Set operations, Sub Queries, Joins; and PL/SQL programs to implement Exceptions, Cursors, Stored Functions, Views, Sequences, Locks and Triggers.

**Course Objectives:**

The objective of conducting this lab is to enable the students to:

1. Demonstrate practical knowledge on creation and alteration of tables, insertion and Querying of data and Analyse and evaluate the databases using SQL DML/DDC commands.
2. Write SQL Queries to implement a Database Schema for the given Database.
3. Design Simple Database using a Tool and Implement it using SQL.
4. Implement DDL and DML commands in SQL and PL/SQL, ORACLE to manage data in databases.
5. Programming PL/SQL including procedures, functions, cursors and triggers.

**Experiments:****1. Practice of DDL and DML Commands and Exercise on DDL and DML Commands**

- a. Create student table with the following structure name varchar2(10), rollno varchar2(5).
- b. Add branch column to the student table.
- c. Increase the size of rollno column from 5 to 10.
- d. Add marks1, marks2, marks3, dob columns to the student table.
- e. Insert at least 5 rows to the student table
- f. Display name and rollno of all the students.
- g. Add total column to the student table.
- h. calculate the total of every student.
- i. Display the details of 'cse' students.
- j. Display the structure of student table.
- k. Display all rows and columns from student table.

- l. Display the name and total obtained by every student.
- m. Display the details of students who got above 250 marks.
- n. Add average column to student table.
- o. Find out the average marks for every student.
- p. Display the details of students who got above 80%.
- q. Modify the branch values from 'CSE' to 'ECE'.
- r. Display the details of students whose dob is 12-05-1991.
- s. Remove details of ECE students.
- t. Remove dob column from student table.
- u. Remove all the records from student table.

## **2. Practice of SQL Operators (Arithmetic, Logical, Relational, Between and, is null and like operator)**

### **3. Exercise on SQL Operators**

- a. Create employee table with the following structure empname varchar2(15), empno number(3), salary number(5), deptno number(3).
- b. Insert at least 5 rows into employee table.
- c. Display the details of employee table.
- d. Eliminate duplicate rows from employee table if any.
- e. Eliminate duplicate names in employee table.
- f. Display the details of employee table in salary ascending order.
- g. Display the details of employee table in empname alphabetic order.
- h. Display the details of employee table in salary descending order.
- i. Create table emp1 which is having same information as employee.
- j. Create a table emp2 from emp1 without getting the records of emp1 table.
- k. Display the details of employee table whose salary is  $\leq 8000$ .
- l. Increment the salaries of employee by 20%.
- m. Display empname, empno who is working in deptno 10;
- n. Display empname, empno who is not working in deptno 30.

- o. Display the details of employee table whose salary in between 8000 and 20000.
- p. Add a column commission number (4) to employee table.
- q. Update the employee table such that, if salary is > 10000 then commission is 2000.
- r. Update the employee table such that, if salary is between 5000 and 10000 then commission is 1000.
- s. Display the details of employee whose name contain 3<sup>rd</sup> letter as 'r'.
- t. Display the details of employee whose name contains exactly 7 characters.
- u. Display the details of employee whose commission is null.
- v. Display the details of employee who is working in deptno 20 and 30.
- w. Display the details of employee whose commission is not null.
- x. Display the details of employee whose salary is 20000 or deptno is 20.
- y. Display the details of employee whose salary is 25000 and deptno is 10.
- z. Display the details of employee whose salary is not 25000.
- aa. Create table emp3 which contains the details of employee table in ascending order of their salary.

**4. Practice of different Integrity Constraints like Not null, Check, Primary Key , Unique Key and Referential Integrity (Foreign Key) Constraints.**

**5. Exercise on Integrity Constraints**

- a. Create emp table with the following structure empno number(6), empname varchar2(10), job char(10), deptno number(4), salary number(5), comm number(3). Job values are Sales executive, Analyst, Manager, Clerk. Allow null values for all columns except empno and deptno.
- b. Add a column phoneno number(10) to emp table.
- c. Modify the column width of Job field of emp table.
- d. Create dept table with the following structure deptno number(4) primary key, deptname char(15), location char(15). deptname values (Sales, Marketing, Production).
- e. Add constraints to emp table such that empno as primary key and deptno as foreign key.
- f. Add constraints to emp table to check the value of empno greater than 100 while entering values.

- g. Add constraints to emp table such that the default salary is 5000 otherwise the values entered by the user.
- h. Add columns DOJ and DOB to emp table.
- i. Insert at least 5 rows into dept table.
- j. Insert at least 10 rows into emp table.
- k. Update the emp table set the value of comm as 999.
- l. Create a table emp1 with the same structure as emp and insert rows into the table using select class.
- m. Delete those rows from emp that have deptno value as 10.
- n. Display unique deptno values from emp table.
- o. Display maximum and minimum salary of emp table and display the output under heading “Minimum Salary” and “Maximum Salary” respectively.
- p. Find out the average salary of emp table.
- q. Find out the 5<sup>th</sup> maximum salary of emp table.
- r. Find out the 3<sup>rd</sup> lowest salary of emp table.
- s. Display the first 3 highest salaries and last 3 lowest salaries from emp table.
- t. Drop the comm column from emp table.
- u. Create a view View1 (empno, deptno, salary) from emp and perform insert, delete and update operations.
- v. Create a view View2 (empno, salary) from emp, (deptno, deptname) from dept and perform insert, delete and update operations.
- w. Drop the view named View2.
- x. Display all emp values from sales or production.
- y. Display the details of employees who joined between 01-01-2020 and 1-06-2021.

**6. Exercise on Relational Algebra operations (Set operations, Different Types of Joins like Natural Join, Equi Join, Theta Join and outer join)**

**7. Practice of SQL Functions(Number, Character, Date , Conversion and Aggregate functions)**

**8. PL/SQL Programs**

- a. Write a PL/SQL program to swap two numbers.
- b. Write a PL/SQL program to find the largest of three numbers.

- c. Write a PL/SQL program to find the total and average of 6 subjects and display the grade.
- d. Write a PL/SQL program to find the sum of digits in a given number.
- e. Write a PL/SQL program to display the number in reverse order.
- f. Write a PL / SQL program to check whether the given number is perfect number or not.
- g. Write a PL/SQL program to find the factorial of a given number.
- h. Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius and the corresponding values of calculated area in an empty table named areas, consisting of two columns radius and area.

## 9. Triggers and Cursors

### 10. CASE STUDY: GENERAL HOSPITAL

- a. A General Hospital consists of a number of specialized wards (such as Maternity, Pediatric, Oncology, etc). Each ward hosts a number of patients, who were admitted on the recommendation of their own GP and confirmed by a consultant employed by the Hospital. On admission, the personal details of every patient are recorded. A separate register is to be held to store the information of the tests undertaken and the results of a prescribed treatment. A number of tests may be conducted for each patient. Each patient is assigned to one leading consultant but may be examined by another doctor, if required. Doctors are specialists in some branch of medicine and may be leading consultants for a number of patients, not necessarily from the same ward.

For the above case study, do the following.

Analyse the data required.

Normalize the attributes.

Create the logical data model using E-R diagrams.

### b. CASE STUDY: STUDENT PROGRESS MONITORING SYSTEM

A database is to be designed for a university to monitor students' progress throughout their course of study. The students are reading for a degree (such as BA, BA (Hons) M.Sc., etc) within the framework of the modular system. The college provides a number of modules, each being characterized by its code, title, credit value, module leader, teaching staff and the department they come from. A module is coordinated by a module leader who shares teaching duties with one or more lecturers. A lecturer may teach (and be a module leader for) more than one module. Students are free to choose any module they wish but the following rules must be observed: Some modules require pre- requisites modules and some degree programmes have compulsory modules. The database is also to contain some information about students including their numbers, names, addresses, degrees they read for, and their past performance i.e. modules taken and examination results.

For the above case study, do the following:

Analyse the data required.

Normalize the attributes.

Create the logical data model i.e., ER diagrams

**Course Outcomes:**

After completion of the course - the student will be able to:

1. Design database for any real-world problem
2. Define SQL queries
3. Decide the constraints
4. Investigate for data inconsistency
5. Implement PL/SQL programs

**References:**

1. Ivan Bayross, "SQL, PL/SQL Programming", 2/e, 2011, BPB Publications, New Delhi, India.
2. Satish Ansari, "Oracle Database 11g: Hands-on SQL and PL/SQL", PHI Publishers, 2010.

**Course Description:**

This course provides an in-depth exploration of essential concepts and practical skills in cloud computing and DevOps methodologies. Students will learn the fundamentals of provisioning, deploying, and managing cloud-based infrastructure, as well as implementing DevOps practices for efficient software development and deployment. Topics include virtualization, containerization, serverless computing, CI/CD pipelines, and cloud migration strategies.

**Course Objectives:**

The students will be able to learn:

1. Understand the principles and benefits of cloud computing and DevOps methodologies.
2. Gain proficiency in provisioning and configuring virtual machines on cloud platforms.
3. Develop skills in containerization technologies and orchestration with Kubernetes.
4. Learn to build and deploy serverless functions using cloud provider services.
5. Explore continuous integration and continuous deployment (CI/CD) practices for automating software delivery pipelines.

**List of Experiments:**

1. Virtual Machine Provisioning and Configuration
2. Containerization and Deployment with Kubernetes
3. Serverless Function Development and Deployment
4. Managed Database Setup and Optimization
5. Data Processing Pipeline Implementation
6. Network Configuration and Security Setup
7. Load Testing and Performance Optimization
8. Continuous Integration/Continuous Deployment (CI/CD) Pipeline Setup
9. Machine Learning Model Training and Deployment
10. Mini Project: Cloud Migration Project Analysis

**Course Outcomes:**

After completion of the course, students will be able to

1. Articulate the key concepts and advantages of cloud computing and DevOps methodologies.
2. Demonstrate the ability to provision, configure, and manage virtual machines in cloud environments.
3. Apply containerization techniques and Kubernetes orchestration to deploy and manage containerized applications.
4. Design and implement serverless functions for various use cases, leveraging cloud provider platforms.
5. Develop and optimize CI/CD pipelines to automate software delivery processes, enhancing efficiency and reliability in software development and deployment.

**Reference Books:**

1. Google Cloud Computing Foundations, Google Skill Development, [https://www.cloudskillsboost.google/course\\_templates/153](https://www.cloudskillsboost.google/course_templates/153).
2. Thomas Erl, Ricardo Puttini, and Zaigham Mahmood, "Cloud Computing: Concepts, Technology & Architecture", 1st Edition, PHI, 2023.
3. JJ Geewax , "Google Cloud Platform in Action", 1st edition, Manning Publications, 2018.
4. Borko Furht and Armando Escalante, "Handbook of Cloud Computing", Springer-Verlag New York Inc, 2014
5. Srinivasan, "Cloud Computing: A Practical Approach for Learning and Implementation", 1st edition, Pearson Education India, 2014.