CO 4	Formulate SQL queries using constraints and set comparison operators.
CO 5	Apply the normalization techniques for development of application software to realistic problems.
CO 6	Develop PL/SQL programs using triggers, procedures
CO 7	Ability to design and implement given case study.

Course Nature		**Practical**		
Assessment Method				
Assessment Tool	Experiments	Record	Viva-Voce/ Quiz/MCQ/Lab project	Total
Weightage (%)	25%	5%	10%	40%
End Semester Examination weightage (%)				60%

ENGINEERING SECOND YEAR: SEMESTER-2

Course code	Course name	Course Category	L-T-P	Credits
20BM2202	**Introduction to Operation Research**	**HSC**	**3-0-0**	**3**

Course Learning Objective:

1. The objective of this course is to provide the exposure to the major tools and techniques of Operations Research.
2. To create awareness and appreciation about the applications of Operations Research in the functional areas of management.
3. To provide a formal quantitative approach to problem solving and an intuition about situations where such an approach is appropriate.
4. To introduce some widely- used mathematical models.
5. The understanding of these models will allow the students to communicate with persons who run them and to evaluate the results they present.
6. To provide a tool that the students can use to solve management problems

Course Contents:

Unit I: (6 hours)

Decision making, Development of OR, An overview and scope of Operations Research Application of OR

Unit II: (9 hours)

Linear Programming Problems (LPP), Introduction to Linear Programming (LP) Illustration of LP Problems. Formulation exercises on LP Problem, Graphical Method of solving LPP Simplex Method, Unboundedness, Multiple Optimum Solutions, Degeneracy and Cycling Problems.

UNIT III: (9 hours)

Artificial Variables : Big-M Method, Sensitivity Analysis, Duality Problems, Economic Interpretation of Simplex Tableau Computer Software for Solving LPP.

UNIT IV: **(9 hours)**

Formulation of Transportation Problems**,** Sensitivity Analysis in Transportation Problems**,** Assignment Problems.

UNIT V: **(7 hours)**

Elements of queuing models, Poisson arrival and exponential service time distributions, M/M/1 Queue; Finite population models. Queuing cost models, Applications.

UNIT VI: **(5 hours)**

Introduction of Costs, Deterministic and Stochastic models.

Learning Recourses:

Text Book:

1. Taha H.A., *Operation Research- An Introduction*, PHI, (2008)

Reference Books:

7. Ravindran, Phillips & Solberg, *Operations research*, John Wiley, Singapore, (2007)
8. Richard Levin & David Rubin, *Quantitative approaches to Management*, Mc GrawHill International, (1992).
9. Hillier & Lieberman, *Operation Research*, Addison Wesley, (1974)
10. Hadley G., *Linear Programming*, Addison-Wesley, (1962).

Web resources:

1. Introduction to Operation Research, Prof. G.S, Srinivasan, IIT Madras
https://nptel.ac.in/courses/110106062/

Course outcomes: At the end of the course, the student will be able to

CO 1	A student will be able to understand basic of operation research and its multipurpose uses
CO 2	It will provide application facilities of concepts.
CO 3	Students can be able to explore different uses of linear programming with its advance technique.
CO 4	It will enable to take decision regarding assignment, transportation and queuing related issues
CO 5	This will provide alternative techniques with its effective uses
CO 6	It will have a overall impact on the decision taking related to various type of real time issues.

For Theory courses only:

Course Nature		**Theory**		
Assessment Method				
Assessment	Weekly tests	Monthly tests	End Semester Test	Total

Tool				
Weightage (%)	10%	30%	60%	100%

****

Course code	Course name	Course Category	L-T-P	Credits
20CS2201	**COMPUTER ORGANIZATION AND ARCHITECTURE**	**PCC**	**3-0-0**	**3**

Course Learning Objectives:

1. To conceptualize the basics of organizational and architectural issues of a digital computer.
2. To analyze performance issues in processor and memory design of a digital computer.
3. To understand various data transfer techniques in digital computer.
4. To analyze processor performance improvement using instruction level parallelism

Course content:

UNIT-I

Basic Functional blocks of a computer: CPU, memory, input -output subsystems, control unit.

Data Representation: Number systems, signed number representation, fixed and floating point representations, character representation.

UNIT-II

ALU: Computer Integer Arithmetic: addition, subtraction, multiplication, division, floating point arithmetic: Addition, subtraction, multiplication, division.

Instruction set architecture of a CPU registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set. RISC and CISC architecture. Case study instruction sets of some common CPUs.

UNIT-III

CPU control unit design: Introduction to CPU design, Processor Organization, Execution of Complete Execution, Design of Control Unit: hardwired and micro-programmed control, Case study design of a simple hypothetical CPU.

UNIT-IV

Memory system design: Concept of memory: Memory hierarchy, SRAM vs DRAM ,Internal organization of memory chips , cache memory: Mapping functions, replacement algorithms, Memory management, virtual memory.

UNIT-V

Input -output subsystems, I/O transfers: programmed I/O, interrupt driven and DMA.

I/O Buses, Peripheral devices and their characteristics, Disk Performance

UNIT-VI

Performance enhancement techniques: Pipelining: Basic concepts of pipelining,

Through put and speedup, pipeline hazards.

Parallel processing: Introduction to parallel processing, Introduction to Network, Cache coherence

Text Books:

V. C. Hamacher, Z. G. Vranesic and S. G. Zaky, *"Computer Organization,"* 5/e, McGraw Hill, 2002.

William Stallings, *"Computer Organization and Architecture"*: Designing for Performance, 8/e, Pearson Education India. 2010.

Morris Mano, " Computer System Architecture", Pearson Education India, Third edition.

References:

A. S. Tanenbaum, "*Structured Computer Organization"*, 5/e, Prentice Hall of India, 2009.
D. A. Patterson and J. L. Hennessy, *"Computer Organization and Design,"* 4/e, Morgan Kaufmann, 2008.
J. L. Hennessy and D. A. Patterson," *Computer Architecture: A Quantitative Approach",*4/e, Morgan Kaufmann, 2006.
D. V. Hall, "*Microprocessors and Interfacing"*, 2/e, McGraw Hall, 2006
*" 8086 Assembler Tutorial for Beginners "*By Prof. Emerson GiovaniCarati.

Web referneces:

https://en.wikibooks.org/wiki/IB/Group_4/Computer_Science/Computer_Organisation
http://www.cs.uwm.edu/classes/cs458/Lecture/HTML/ch05.html
http://www.cse.iitm.ac.in/~vplab/courses/comp_org.htm
OA IIT-Guwahati Pdf

Course outcomes: At the end of the course, the student will be able to

CO1	Understand the basic organization of computer and different instruction formats and addressing modes.
CO2	Analyze the concept of pipelining, segment registers and pin diagram of CPU.
CO3	Understand and analyze various issues related to memory hierarchy.
CO4	Evaluate various modes of data transfer between CPU and I/O devices.
CO5	Examine various inter connection structures of multi processors.

For Theory courses only:

Course Nature		**Theory**		
Assessment Method				
Assessment Tool	Weekly tests	Monthly tests	End Semester Test	Total
Weightage (%)	10%	30%	60%	100%

****

Course code	Course name	Course Category	L-T-P	Credits
20CS2202	Data Science with Python	PCC	3-0-0	3

Unit – I : Python Basics for Data Science

Introduction to Python, Types, Expressions & Variables, String Operations, Lists & Tuples, Sets, Dictionaries, Conditions & Branching, Loops, Functions, Objects & Classes

Unit – II: Working with Data in Python

File Operations, Regular Expressions, Pandas, NumPys, Web Scraping

Unit – III: Data Processing

Importing DataSets; Cleaning & Preparing Data – Handling Missing Values, Data Formatting, Binning;Summarizing the Data Frame – Descriptive Statistics, Grouping, ANOVA, Correlation

Unit – IV: Data Analysis

Model Development : Simple & Multiple Linear Regression, Model Evaluation using Visualization, Polynomial Regression; Model Evaluation – Overfitting, Underfitting, Model Selection, Ridge Regression, Model Refinement

Unit – V: Data Visualization

Introduction to Visualization Tools – Matplotlib, Line Plots, Area Plots, Histograms, Bar Charts, Pie Charts, Box Plots, Scatter Plots, Bubble Plots; Advance Visualization Tools – Waffle Charts, Word Clouds, Seaborn and Regression Plots; Creating Maps & Visualizing Geospatial Data – Folium, Maps with Markers, Choropleth Maps

Unit – VI: Machine Learning using Python

Introduction to Machine Learning – Supervised vs Unsupervised Learning, Python Libraries for Machine Learning; Regression; Classification; Unsupervised Learning; Recommender Systems

Learning resources:

Text Book:

1) Jake VanderPlas, Python Data Science Handbook - Essential Tools for Working with Data, o'reilly publications.

Online Course Reference:

https://www.edx.org/professional-certificate/python-data-science

****

Course Code	Course Name	Course Category	L-T-P	Credits
20CS2204	Compiler Design	PCC	3-0-0	3

Course Learning Objectives:

1. To implement the concept learned in automata theory and languages to the field of Computer Science.
2. Analyze the basic steps involved in converting a source language to target code.
3. Understands the concepts of parsers and can write solutions for various grammars by using tools, and also analyzes different storage techniques, error recovery strategies
4. Gain the knowledge to write a compiler program or can able to build a compiler.

Course Content:

Unit I

Introduction to Compilers (6 Contact hours)

Introduction to compilers, Phases of compiler, Lexical Analyzer, The role of the lexical analyzer, input buffering, specification of tokens, Recognition of tokens.

Unit II

Syntax Analysis -I (9 Contact hours)

Role of the parser, writing grammars and context free grammars, Top down parsing, Brute-force approach, Recursive descent parsing, Predictive parsing, FIRST and FOLLOW constructs.

Unit III

Syntax Analysis -II (8 Contact hours)

Bottom-up parsing, shift-reduce parsing, operator precedence parsing, LR parsers, SLR parser, canonical LR parser, LALR parser.

Unit IV

Semantic Analysis (8 Contact hours)

Syntax directed translations, applications of syntax directed translations, Syntax directed definitions, construction of syntax tree, Bottom-up evaluation of S-attributed definitions, L-attributed definitions.

Unit V

Intermediate Code Generation and Code Optimization (8 Contact hours)

Intermediate languages, Declarations, Assignment statements, Boolean Expressions, case statements, back patching, Procedure calls, Principal sources of optimization, optimization of basic blocks, DAG representation of basic blocks, flow graphs.

Unit VI

Code generation (6 Contact hours)

Issues in the design of code generator, the target machine, run time storage management, peephole optimization.

Learning resources

Text book:

1. Alfred V Aho, Monica S Lam, Ravi Sethi, Jeffrey D Ullman, *"Compilers: Principles Techniques & Tools"*, Pearson Education,2nd Edition 2013.

Reference Books:

1. Kenneth C Louden, *"Compiler Construction: Principles and Practice*", Cengage Learning. Lex &Yacc, John R Levine, Oreilly Publishers.
2. Keith D Cooper & Linda Tarezon*, "Engineering a Compiler*", Morgan Kafman, Second edition. Lex &Yacc, John R Levine, Tony Mason, Doug Brown, Shroff Publishers.
3. Muchnik, "*Advanced Compiler Design and Implementation*", Kauffman(1998)

Course outcomes:

CO	Identify the basic concepts needed for the development of a compiler

1	
CO 2	Analyze the various phases and Tools of a Compiler
CO 3	Describe the differences between Top-Down and Bottom-Up Parsers and apply parsing methods for various grammars.
CO 4	Compare and Contrast Symbol table organization for Block Structured and Non-Block Structured languages.
CO 5	Analyze the concepts involved in Intermediate, Code Generation and Code Optimization Process.
CO 6	Recognize the various types of errors and error recovery strategies in phases of Compilation

Assessment Method:

Course Nature		**Theory**		
Assessment Method				
Assessment Tool	Weekly tests (In semester)	Monthly tests (In semester)	End Semester Test	Total
Weightage (%)	10%	30%	60%	100%

Course Code	Course Name	Course Category	L-T-P	Credits
20CS2203	Web Technologies	PCC	3-0-0	3

Course Learning Objectives:

1. To demonstrate basic skills in analysing the usability of a web site.
2. To identify how to plan user research related to web design.
3. To Learn how to design, add client side script and publish web page
4. To Learn about server side programming and deploy the app into a server
5. To Learn about storing the data into SQL and NoSQL
6. To Learn about Front-End Web UI Frameworks and GIT repository Tools
7. To learn the language of the Web: jQuery Frontend design and Bootstrap
8. To learn the language of Web: NodeJS

UNIT I (10 Contact hours)

Introduction to Web World: Recap on HTML, inserting Frames and frame sets, inserting hyperlinks, lists, tables and images,

JavaScript : Client side scripting with JavaScript, variables, functions, conditions, loops and repetition, Pop up boxes, objects,HTML DOM and web, Browser environments, form validation, Events and Event Listeners

UNIT II (10 Contact hours)

Server Programming: PHP basics: PHP Syntax, variables, constants, Data Types, Strings, Conditional and Control Structures. PHP GET and POST. PHP Advanced: include files, File system, parsing directories, file upload and download, Sessions, Form handling, JSON Parsing

UNIT-III (10 Contact hours)

Database Connectivity: Introduction to SQL: Connect, create database, create table, insert, prepared statements. Use of NoSQL: Introduction to NoSQL, Difference between SQL and NoSQL, Types of NoSQL Databases, Query mechanism tools for NoSQL.

Authentication: Google OAuth: Basic Steps. Access to Google APIs: For Server-side Web apps, for Java Script Web apps, for Mobile & Desktop apps

UNIT IV (10 Contact hours)

Front-End Web UI Frameworks and Tools: Bootstrap, Full-Stack Web Development, Setting up Git, Basic Git commands, Online Git Repositories, Node.js and NPM, Front-end Web UI Frameworks.

jQuery: Introduction, Selectors, Attributes, Event Handlers, Style Methods, Traversing the DOM, Effects, and Introduction to jQuery Plugins

UNIT V **(7Contact hours)**

Bootstrap: Introduction to Bootstrap, Responsive Design, Bootstrap Grid system, Navigation and Navigation Bar, Icon Fonts, User Inputs, Bootstrap CSS Components, Bootstrap and JavaScript Components, Bootstrap and JQuery, Building and Deployment, NPM Scripts, Task Runners

UNIT VI **(7Contact hours)**

NodeJs: Introduction, Environment Setup, First Application, REPL Terminal, Native Package Manager(NPM), Web Module, Express Framework, RESTFul API

Learning resources:

Text Books

1. Ralph Moseley and M. T. Savaliya, WileyIndia "*Developing Web Applications*"
2. Jeffrey C.Jackson,"*Web Technologies--A Computer Science Perspective*", PearsonEducation,
3. Dreamtech Press "*Web TechnologiesBlack Book*," HTML 5,
4. Web Technologies: HTML, JAVASCRIPT, PHP, JAVA, JSP, XML and AJAX, Black Book Kindle Edition by Kogent Learning Solutions Inc. (Author)
5. O'Reilly - Head First Servlets and JSP, 2nd Edition
6. Node.js Web Development: Create real-time server-side applications with this practical, step-by-step guide, 3rd Edition
7. Bootstrap: Responsive Web Development Book by Jake Spurlock

Reference Books

1. Joel Sklar, Cengage *" Web Design*", Learning
2. Robert. W. Sebesta, "*Programming the World Wide Web*", Fourth Edition, PearsonEducation
3. P.J. Deitel& H.M. Deitel "*Internet and World Wide Web How to program*", Pearson

Web Resources

https://www.w3schools.com/
https://www.tutorialspoint.com/web_development_tutorials.htm
https://html.com/
https://www.coursera.org/learn/bootstrap-4
https://www.tutorialspoint.com/jquery/index.htm
https://www.tutorialspoint.com/nodejs/
http://www.ntu.edu.sg/home/ehchua/programming/java/javaservlets.html
http://wiki.lib.sun.ac.za/images/0/07/Bootstrap-tutorial.pdf
https://media.readthedocs.org/pdf/htmlguide/latest/htmlguide.pdf

Course outcomes: At the end of the course, the student will be able to

CO 1	Learn how to design, add client side script and publish web page

CO 2	Learn how to write server side programming and deploy the app into a server.
CO 3	Learn how to store data into database and NoSQL.
CO 4	Learn about Front-End Web UI Frameworks and GIT repository Tools.
CO 5	Learn about responsive Web design.
CO 6	Learn about Package manager and Web modules.

****

Course code	Course Name	Course Category	L-T-P	Credits
20CS2281	**Computer Organization and Architecture Laboratory**	**PCC**	**0-0-3**	**1.5**

Course Learning Objective

1. To expose the students to the various key aspects of Computer Organization & Architecture.
2. To aquaint with various registers in the CPU and understand about the assembly language programming.

List of Experiments:

Lab No 1.a) Verification of Logic gates
b) Assembly language program to find largest number in an Array.

Lab No 2. a) Verification of Full-Adder and Full-Subtractor
b) Assembly language program to find smallest number in an array.

Lab No 3. a)Verification of Ripple Carry Adder and Carry-look-ahead adder.
b) Assembly language program for adding to two arrays

Lab No 4. a) Combinational Multipliers
b) Assembly language program to separate even and odd numbers from an array.

Lab No 5. a) Booth's Multiplier
b) Assembly language rogram to find prime numbers between a given range

Lab No 6. a) Wallace Tree Adder
b) Assembly language program to find factorial of the given number.

Lab No 7. a) Arithmetic Logic Unit
b) Assembly language program to find LCM.

Lab No 8. a) Verification of Registers and Counters
b) Assembly language program to find GCD.

Lab No 9. a) Memory Design
b) Assembly language program to search an element using linear search.

Lab No 10. a) Direct Mapped cache Design
b) Assembly language program to search an element using binary search.

Lab No 11. a) Associative cache Design

b) Assembly language program to sort numbers using bubble sort.

Course Outcomes

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

CO 1	Understand the basic logic gates
CO 2	Understand the full adder and full subtractor
CO 3	**Ripple Carry Adder** examine the behavior of the working module to understand how the carry ripples through the adder stages to design a ripple carry adder using full adders to mimic the behavior of the working module . **Carry Lookahead Adder** understand the behaviour of carry lookahead adder understand the concept of reducing computation time with respect of ripple carry adder by using carry generate and propagate functions
CO4	**Combinational Multipliers** understand the behaviour of combinational multiplier . understand the scheme implemented for the multiplication. it can be designed by unrolling the multiplier loop instead of handling the carry out of partial product summation bit,the carry out can be sent to the next bit of the next step this scheme of handling the carry is called *carry save addition*
CO 5	**Booth's Multiplier** Understand the behaviour of Booth's multiplication**.** Design Booth's multiplier with a controller and a datapath. This will also help in the learning of control unit design as a finite state machine Understand the advantages of Booth's multiplier It can handle signed integers in 2's complement notion It decreases the number of addition and subtraction It requires less hardware than combinational multiplier It is faster than straightforward sequential multiplier
CO 6	**Wallace Tree Adder** Understand the behaviour of wallace tree**.** understand the concept of reducing gate delay by using tree of adders instead of using cascaded full adders
CO 7	**Arithmetic Logic Unit** Understand the behaviourof arithmetic logic unit. Design an arithmetic logic unit for given parameter**.**
CO8	**Registers** to understand the shifting of data to examine the behavior of different modes of data input and data output(serial-in

	serial-out, serial-in parallel-out, parallel-in serial out,parallel-in parallel-out) to make use of shift register in data transfer developing skills in the designing and testing of sequential logic circuits developing skills in analysing timing signals. **Counters** understand the concept of counting upto certain limiting value and returning back to the start state from final state understand the generation of timing sequences to control operations in a digital system develop skills in the design and testing of counters for given timing sequences develop skills in generating timing signals .
CO 9	**Memory Design** Understand the **behavior of memory.** **D**esign memory for given parameter**.**
CO 10	**Direct Mapped Cache Design** Understand the behavior of direct mapped cache from working module Design a direct mapped cache for given parameters.
CO 11	**Associative Cache Design** Understand the behavior of associative cache. Designs a associative cache for given parameters**.**
Understand and develop Assembly Language Programs	

Assessment Method

Assessment Tool	Experiments	Record/Viva-Voce/ Quiz/MCQ/Lab project	Total
Weightage (%)	25%	15%	40%
End Semester Examination weightage (%)			60%

****

Course code	Course name	Course Category	L-T-P	Credits
20CS2282	**Data Science using Python Lab**		**0-0-3**	**1.5**

Lab:

Experiment 1:

a) **Python Basics:** Your first program, Types Expressions and Variables String Operations

b) **Python Data Structures:** Lists and Tuples Sets,and Dictionaries

c) **Python Programming Fundamentals:** Conditions and Branching Loops, Functions, Objects and Classes

d) **Working with Data in Python:** Reading files with open, Writing files with open, Loading data with Pandas, Working with and Saving data with Pandas

e) **Working with Numpy Arrays:** Numpy 1d Arrays, Numpy 2d Arrays

Experimen 2:

a) **Importing Datasets:** Learning Objectives, Understanding the Domain, Understanding the Dataset, Python package for data science, Importing and Exporting Data in Python, Basic Insights from Datasets

b) **Cleaning and Preparing the Data:** Identify and Handle Missing Values, Data Formatting, Data Normalization Sets, Binning, Indicator variables

c) **Model Development:** Simple and Multiple Linear Regression, Model EvaluationUsingVisualization, Polynomial Regression and Pipelines, R-squared and MSE for In-Sample Evaluation, Prediction and Decision Making

d) **Summarizing the Data Frame:** Descriptive Statistics, Basic of Grouping, ANOVA, Correlation, More on Correlation

e) **Model Evaluation:** Model Evaluation, Over-fitting, Under-fitting and Model Selection, Ridge Regression, Grid Search, Model Refinement

Experiment 3:

a) **Introduction to Visualization Tools:** Introduction to Data Visualization,Introduction to Matplotlib, Basic Plotting with Matplotlib,Dataset on Immigration to Canada, Line Plots

b) **Basic Visualization Tools:** Area Plots,Histograms,Bar Charts

c) **Specialized Visualization Tools:** Pie Charts, Box Plots, Scatter Plots, Bubble Plots

d) **Advanced Visualization Tools:** Waffle Charts, Word Clouds,Seaborn and Regression Plots

Expeiment 4:

a) **Introduction to Machine Learning:** Applications of Machine Learning, Supervised vs Unsupervised Learning, Python libraries suitable for Machine Learning

b) **Regression:** Linear Regression, Non-linear Regression, Model evaluation methods

c) **Classification:** K-Nearest Neighbour, Decision Trees, Logistic Regression, Support Vector Machines,Model Evaluation

d) **Unsupervised Learning:**K-Means Clustering,Hierarchical Clusterin, Density-Based Clustering

Assessment Method

Assessment Tool	Experiments	Report/Viva-Voce/ Quiz/MCQ/Lab project	Total
Weightage (%)	25%	15%	40%
End Semester Examination weightage (%)			60%

****

Course code	Course name	Course Category	L-T-P	Credits
20CS2283	**Web Technologies Lab**	**PCC**	**0-0-3**	**1.5**

Course Learning Objective:

The course will enable the students to:

1. Demonstrate the ability to retrieve data from a database and present it in a web page.
2. Create web pages that meet accessibility needs of those with physical disabilities and apply the effects of CSS in web page creation.
3. Create effective scripts using JavaScript and jQuery to enhance the end user experience.
4. Demonstrate knowledge of introductory programming concepts.
5. Test, debug, and deploy web pages containing JavaScript and jQuery

6. learn to create pages common to all web applications, and implement the most frequently used components and classes provided by Bootstrap
7. Understand the JavaScript and technical concepts behind Node JS
8. Understand the Servlet programming and deploying application in Web server

List of Experiments:

1. Install and configure the IDE
2. Incorporating JavaScript on an HTML page, and how to link to an external .js file
3. Comparing JavaScript with jQuery for same tasks
4. Using major methods/events in jQuery
5. Using Plugins and local data storage
6. Implement Bootstrap in existing web sites
7. Common Bootstrap components and use Bootstrap themes
8. Setup a Node.js project using npm
9. Use the Node.js core modules

Course Outcomes

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

CO 1	and configure the IDE
CO 2	orating JavaScript on an HTML page, and how to link to an external .js file
CO 3	Comparing JavaScript with jQuery for same tasks
CO 4	major methods/events in jQuery
CO 5	Plugins and local data storage
CO 6	Implement Bootstrap in existing web sites
CO 7	ıon Bootstrap components and use Bootstrap themes
CO 8	a Node.js project using npm
CO 9	e Node.js core modules
CO 10	ng Servlet application and deploying application in web server

Assessment Method

Assessment Tool	Experiments	Report/Viva-Voce/ Quiz/MCQ/Lab project	Total
Weightage (%)	25%	15%	40%
End Semester Examination weightage (%)			60%

****

ENGINEERING THIRD YEAR : SEMSTER-I

Course Code	Course Name	Course Category	L-T-P	Credits
20CS3101	**Operating System**	**PCC**	**3-0-0**	**3**

Course Learning Objectives

1. To learn the fundamentals of Operating Systems.
2. To learn the mechanisms of OS to handle processes and threads and their communication