### II YEAR I SEMESTER

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CS301ES</td>
<td>Analog and Digital Electronics</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>CS302PC</td>
<td>Data Structures</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>MA303BS</td>
<td>Computer Oriented Statistical Methods</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>CS304PC</td>
<td>Computer Organization and Architecture</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>CS311PC</td>
<td>Python Programming</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>CS306ES</td>
<td>Analog and Digital Electronics Lab</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>CS307PC</td>
<td>Data Structures Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>8</td>
<td>CS308PC</td>
<td>IT Workshop Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>9</td>
<td>CS312PC</td>
<td>Python Programming Lab</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>*MC309</td>
<td>Gender Sensitization Lab</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td><strong>Total Credits</strong></td>
<td></td>
<td><strong>14</strong></td>
<td><strong>2</strong></td>
<td><strong>12</strong></td>
<td><strong>21</strong></td>
</tr>
</tbody>
</table>

### II YEAR II SEMESTER

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CS401PC</td>
<td>Discrete Mathematics</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>SM402MS</td>
<td>Business Economics &amp; Financial Analysis</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>CS403PC</td>
<td>Operating Systems</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>CS418PC</td>
<td>Principles of Software Engineering</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>CS412PC</td>
<td>Object Oriented Programming using Java</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>CS406PC</td>
<td>Operating Systems Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>7</td>
<td>CS419PC</td>
<td>Software Engineering Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>8</td>
<td>CS408PC</td>
<td>Java Programming Lab</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>*MC409</td>
<td>Constitution of India</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td><strong>Total Credits</strong></td>
<td></td>
<td><strong>18</strong></td>
<td><strong>2</strong></td>
<td><strong>8</strong></td>
<td><strong>21</strong></td>
</tr>
</tbody>
</table>

**JAYWALRAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**B.Tech. in COMPUTER ENGINEERING (SOFTWARE ENGINEERING)**

**COURSE STRUCTURE & SYLLABUS (R18)**

Applicable From 2020-21 Admitted Batch
CS301ES: ANALOG AND DIGITAL ELECTRONICS

B.Tech. II Year I Sem.  

Course Objectives:
- To introduce components such as diodes, BJTs and FETs.
- To know the applications of components.
- To give understanding of various types of amplifier circuits
- To learn basic techniques for the design of digital circuits and fundamental concepts used in the design of digital systems.
- To understand the concepts of combinational logic circuits and sequential circuits.

Course Outcomes: Upon completion of the Course, the students will be able to:
- Know the characteristics of various components.
- Understand the utilization of components.
- Design and analyze small signal amplifier circuits.
- Learn Postulates of Boolean algebra and to minimize combinational functions
- Design and analyze combinational and sequential circuits
- Know about the logic families and realization of logic gates.

UNIT - I
Diodes and Applications: Junction diode characteristics: Open circuited p-n junction, p-n junction as a rectifier, V-I characteristics, effect of temperature, diode resistance, diffusion capacitance, diode switching times, breakdown diodes, Tunnel diodes, photo diode, LED.
Diode Applications - clipping circuits, comparators, Half wave rectifier, Full wave rectifier, rectifier with capacitor filter.

UNIT - II
BJTs: Transistor characteristics: The junction transistor, transistor as an amplifier, CB, CE, CC configurations, comparison of transistor configurations, the operating point, self-bias or Emitter bias, bias compensation, thermal runaway and stability, transistor at low frequencies, CE amplifier response, gain bandwidth product, Emitter follower, RC coupled amplifier, two cascaded CE and multi stage CE amplifiers.

UNIT - III
FETs and Digital Circuits: FETs: JFET, V-I characteristics, MOSFET, low frequency CS and CD amplifiers, CS and CD amplifiers.
Digital Circuits: Digital (binary) operations of a system, OR gate, AND gate, NOT, EXCLUSIVE OR gate, De Morgan Laws, NAND and NOR DTL gates, modified DTL gates, HTL and TTL gates, output stages, RTL and DCTL, CMOS, Comparison of logic families.

UNIT - IV

UNIT - V
Sequential Logic Circuits: Sequential Circuits, Storage Elements: Latches and flip flops, Analysis of Clocked Sequential Circuits, State Reduction and Assignment, Shift Registers, Ripple Counters, Synchronous Counters, Random-Access Memory, Read-Only Memory.
TEXTBOOKS:

REFERENCE BOOKS:
CS302PC: DATA STRUCTURES

B.Tech. II Year I Sem.  

Prerequisites: A course on “Programming for Problem Solving”.

Course Objectives:
- Exploring basic data structures such as stacks and queues.
- Introduces a variety of data structures such as hash tables, search trees, tries, heaps, graphs.
- Introduces sorting and pattern matching algorithms.

Course Outcomes:
- Ability to select the data structures that efficiently model the information in a problem.
- Ability to assess efficiency trade-offs among different data structure implementations or combinations.
- Implement and know the application of algorithms for sorting and pattern matching.
- Design programs using a variety of data structures, including hash tables, binary and general tree structures, search trees, tries, heaps, graphs, and AVL-trees.

UNIT - I
Introduction to Data Structures, abstract data types, Linear list – singly linked list implementation, insertion, deletion and searching operations on linear list, Stacks-Operations, array and linked representations of stacks, stack applications, Queues-operations, array and linked representations.

UNIT - II
Dictionaries: linear list representation, skip list representation, operations - insertion, deletion and searching. 
Hash Table Representation: hash functions, collision resolution-separate chaining, open addressing-linear probing, quadratic probing, double hashing, rehashing, extendible hashing.

UNIT - III

UNIT - IV
Graphs: Graph Implementation Methods. Graph Traversal Methods. 
Sorting: Heap Sort, External Sorting- Model for external sorting, Merge Sort.

UNIT - V
Pattern Matching and Tries: Pattern matching algorithms-Brute force, the Boyer –Moore algorithm, the Knuth-Morris-Pratt algorithm, Standard Tries, Compressed Tries, Suffix tries.

TEXT BOOKS:

REFERENCE BOOK:
MA303BS: COMPUTER ORIENTED STATISTICAL METHODS

B.Tech. II Year I Sem.  L  T  P  C

Pre-requisites: Mathematics courses of first year of study.

Course Objectives: To learn
- The theory of Probability, and probability distributions of single and multiple random variables
- The sampling theory and testing of hypothesis and making inferences
- Stochastic process and Markov chains.

Course Outcomes: After learning the contents of this paper the student must be able to
- Apply the concepts of probability and distributions to some case studies
- Correlate the material of one unit to the material in other units
- Resolve the potential misconceptions and hazards in each topic of study.

UNIT - I

UNIT - II
Mathematical Expectation: Mean of a Random Variable, Variance and Covariance of Random Variables, Means and Variances of Linear Combinations of Random Variables, Chebyshev’s Theorem.

UNIT - III
Continuous Probability Distributions: Continuous Uniform Distribution, Normal Distribution, Areas under the Normal Curve, Applications of the Normal Distribution, Normal Approximation to the Binomial, Gamma and Exponential Distributions.

UNIT - IV

UNIT - V
TEXT BOOKS:

REFERENCE BOOKS:
CS304PC: COMPUTER ORGANIZATION AND ARCHITECTURE

B.Tech. II Year I Sem. L T P C
3 0 0 3

Co-requisite: A Course on “Digital Logic Design and Microprocessors”.

Course Objectives:
- The purpose of the course is to introduce principles of computer organization and the basic architectural concepts.
- It begins with basic organization, design, and programming of a simple digital computer and introduces simple register transfer language to specify various computer operations.
- Topics include computer arithmetic, instruction set design, microprogrammed control unit, pipelining and vector processing, memory organization and I/O systems, and multiprocessors.

Course Outcomes:
- Understand the basics of instructions sets and their impact on processor design.
- Demonstrate an understanding of the design of the functional units of a digital computer system.
- Evaluate cost performance and design trade-offs in designing and constructing a computer processor including memory.
- Design a pipeline for consistent execution of instructions with minimum hazards.
- Recognize and manipulate representations of numbers stored in digital computers.

UNIT - I
Register Transfer Language and Micro operations: Register Transfer language, Register Transfer, Bus and memory transfers, Arithmetic Micro operations, logic micro operations, shift micro operations, Arithmetic logic shift unit.
Basic Computer Organization and Design: Instruction codes, Computer Registers Computer instructions, Timing and Control, Instruction cycle, Memory Reference Instructions, Input – Output and Interrupt.

UNIT - II
Microprogrammed Control: Control memory, Address sequencing, micro program example, design of control unit.
Central Processing Unit: General Register Organization, Instruction Formats, Addressing modes, Data Transfer and Manipulation, Program Control.

UNIT - III
Data Representation: Data types, Complements, Fixed Point Representation, Floating Point Representation.

UNIT - IV
Memory Organization: Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory.
UNIT - V

Reduced Instruction Set Computer: CISC Characteristics, RISC Characteristics.

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processor.

Multi Processors: Characteristics of Multiprocessors, Interconnection Structures, Interprocessor arbitration, Interprocessor communication and synchronization, Cache Coherence.

TEXT BOOK:

REFERENCE BOOKS:
CS311PC: PYTHON PROGRAMMING

B.Tech. II Year I Sem. L T P C 2 0 0 2

Prerequisites: A course on “Programming for Problem Solving using C”.

Course Objectives:

- Learn Syntax and Semantics and create Functions in Python.
- Handle Strings and Files in Python.
- Understand Lists, Dictionaries and Regular expressions in Python.
- Implement Object Oriented Programming concepts in Python.
- Build Web Services and introduction to Network and Database Programming in Python.

Course Outcomes:

- Examine Python syntax and semantics and be fluent in the use of Python flow control and functions.
- Demonstrate proficiency in handling Strings and File Systems.
- Create, run and manipulate Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions.
- Interpret the concepts of Object-Oriented Programming as used in Python.
- Implement exemplary applications related to Network Programming, Web Services and Databases in Python.

UNIT - I
Python Basics, Objects- Python Objects, Standard Types, Other Built-in Types, Internal Types, Standard Type Operators, Standard Type Built-in Functions, Categorizing the Standard Types, Unsupported Types Numbers - Introduction to Numbers, Integers, Floating Point Real Numbers, Complex Numbers, Operators, Built-in Functions, Related Modules Sequences - Strings, Lists, and Tuples, Mapping and Set Types

UNIT - II

UNIT - III

UNIT - IV
UNIT - V
Database Programming: Introduction, Python Database Application Programmer’s Interface (DB-API), Object Relational Managers (ORMs), Related Modules

TEXT BOOK:

REFERENCE BOOKS:
1. Think Python, Allen Downey, Green Tea Press
2. Introduction to Python, Kenneth A. Lambert, Cengage
4. Learning Python, Mark Lutz, O’Really
CS306ES: ANALOG AND DIGITAL ELECTRONICS LAB

B.Tech. II Year I Sem. L T P C

0 0 2 1

Course Objectives
- To introduce components such as diodes, BJT's and FET's.
- To know the applications of components.
- To give understanding of various types of amplifier circuits.
- To learn basic techniques for the design of digital circuits and fundamental concepts used in the design of digital systems.
- To understand the concepts of combinational logic circuits and sequential circuits.

Course Outcomes: Upon completion of the Course, the students will be able to:
- Know the characteristics of various components.
- Understand the utilization of components.
- Design and analyze small signal amplifier circuits.
- Postulates of Boolean algebra and to minimize combinational functions.
- Design and analyze combinational and sequential circuits.
- Known about the logic families and realization of logic gates.

List of Experiments
1. Full Wave Rectifier with & without filters
2. Common Emitter Amplifier Characteristics
3. Common Base Amplifier Characteristics
4. Common Source amplifier Characteristics
5. Measurement of h-parameters of transistor in CB, CE, CC configurations
6. Input and Output characteristics of FET in CS configuration
7. Realization of Boolean Expressions using Gates
8. Design and realization logic gates using universal gates
9. generation of clock using NAND / NOR gates
10. Design a 4 – bit Adder / Subtractor
11. Design and realization a Synchronous and Asynchronous counter using flip-flops
12. Realization of logic gates using DTL, TTL, ECL, etc.
CS307PC: DATA STRUCTURES LAB

B.Tech. II Year I Sem.  

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Prerequisites: A Course on “Programming for problem solving”.

Course Objectives:
- It covers various concepts of C programming language
- It introduces searching and sorting algorithms
- It provides an understanding of data structures such as stacks and queues.

Course Outcomes:
- Ability to develop C programs for computing and real-life applications using basic elements like control statements, arrays, functions, pointers and strings, and data structures like stacks, queues and linked lists.
- Ability to implement searching and sorting algorithms

List of Experiments
1. Write a program that uses functions to perform the following operations on singly linked list:
   i) Creation  ii) Insertion  iii) Deletion  iv) Traversal
2. Write a program that uses functions to perform the following operations on doubly linked list:
   i) Creation  ii) Insertion  iii) Deletion  iv) Traversal
3. Write a program that uses functions to perform the following operations on circular linked list:
   i) Creation  ii) Insertion  iii) Deletion  iv) Traversal
4. Write a program that implements stack (its operations) using
   i) Arrays  ii) Pointers
5. Write a program that implements Queue (its operations) using
   i) Arrays  ii) Pointers
6. Write a 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
7. Write a program that uses 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
8. Write a program to implement the tree traversal methods.
9. Write a program to implement the graph traversal methods.

TEXT BOOKS:

REFERENCE BOOK:
CS308PC: IT WORKSHOP LAB

B.Tech. II Year I Sem.  

L T P C  0 0 3 1.5

Course Objectives:  
The IT Workshop for engineers is a training lab course spread over 60 hours. The modules include training on PC Hardware, Internet & World Wide Web and Productivity tools including Word, Excel, Power Point and Publisher.

PC Hardware  
introduces the students to a personal computer and its basic peripherals, the process of assembling a personal computer, installation of system software like MS Windows, Linux and the required device drivers. In addition hardware and software level troubleshooting process, tips and tricks would be covered. The students should work on working PC to disassemble and assemble to working condition and install Windows and Linux on the same PC. Students are suggested to work similar tasks in the Laptop scenario wherever possible. Internet & World Wide Web module introduces the different ways of hooking the PC on to the internet from home and workplace and effectively usage of the internet. Usage of web browsers, email, newsgroups and discussion forums would be covered. In addition, awareness of cyber hygiene, i.e., protecting the personal computer from getting infected with the viruses, worms and other cyber attacks would be introduced. Productivity tools module would enable the students in crafting professional word documents, excel spreadsheets, power point presentations and personal web sites using the Microsoft suite of office tools and LaTeX.

PC Hardware  
Task 1: Identify the peripherals of a computer, components in a CPU and its functions. Draw the block diagram of the CPU along with the configuration of each peripheral and submit to your instructor.

Task 2: Every student should disassemble and assemble the PC back to working condition. Lab instructors should verify the work and follow it up with a Viva. Also students need to go through the video which shows the process of assembling a PC. A video would be given as part of the course content.

Task 3: Every student should individually install MS windows on the personal computer. Lab instructor should verify the installation and follow it up with a Viva.

Task 4: Every student should install Linux on the computer. This computer should have windows installed. The system should be configured as dual boot with both windows and Linux. Lab instructors should verify the installation and follow it up with a Viva.

Task 5: Hardware Troubleshooting: Students have to be given a PC which does not boot due to improper assembly or defective peripherals. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva.

Task 6: Software Troubleshooting: Students have to be given a malfunctioning CPU due to system software problems. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva.

Internet & World Wide Web  
Task 1: Orientation & Connectivity Boot Camp: Students should get connected to their Local Area Network and access the Internet. In the process they configure the TCP/IP setting. Finally students
should demonstrate, to the instructor, how to access the websites and email. If there is no internet
connectivity preparations need to be made by the instructors to simulate the WWW on the LAN.

**Task 2: Web Browsers, Surfing the Web:** Students customize their web browsers with the LAN
proxy settings, bookmarks, search toolbars and pop up blockers. Also, plug-ins like Macromedia
Flash and JRE for applets should be configured.

**Task 3: Search Engines & Netiquette:** Students should know what search engines are and how to
use the search engines. A few topics would be given to the students for which they need to search on
Google. This should be demonstrated to the instructors by the student.

**Task 4: Cyber Hygiene:** Students would be exposed to the various threats on the internet and would
be asked to configure their computer to be safe on the internet. They need to first install an antivirus
software, configure their personal firewall and windows update on their computer. Then they need to
customize their browsers to block pop ups, block active x downloads to avoid viruses and/or worms.

**LaTeX and WORD**

**Task 1 – Word Orientation:** The mentor needs to give an overview of LaTeX and Microsoft (MS)
office 2007/ equivalent (FOSS) tool word: Importance of LaTeX and MS office 2007/ equivalent
(FOSS) tool Word as word Processors, Details of the four tasks and features that would be covered in
each, Using LaTeX and word – Accessing, overview of toolbars, saving files, Using help and
resources, rulers, format painter in word.

**Task 2: Using LaTeX and Word** to create project certificate. Features to be covered:- Formatting
Fonts in word, Drop Cap in word, Applying Text effects, Using Character Spacing, Borders and
Colors, Inserting Header and Footer, Using Date and Time option in both LaTeX and Word.

**Task 3: Creating project abstract** Features to be covered:- Formatting Styles, Inserting table, Bullets
and Numbering, Changing Text Direction, Cell alignment, Footnote, Hyperlink, Symbols, Spell Check,
Track Changes.

**Task 4: Creating a Newsletter:** Features to be covered:- Table of Content, Newspaper columns,
Images from files and clipart, Drawing toolbar and Word Art, Formatting Images, Textboxes,
Paragraphs and Mail Merge in word.

**Excel**

**Excel Orientation:** The mentor needs to tell the importance of MS office 2007/ equivalent (FOSS)
tool Excel as a Spreadsheet tool, give the details of the four tasks and features that would be covered in
each. Using Excel – Accessing, overview of toolbars, saving excel files, Using help and resources.

**Task 1: Creating a Scheduler** - Features to be covered: Gridlines, Format Cells, Summation, auto
fill, Formatting Text

**Task 2 : Calculating GPA** - .Features to be covered:- Cell Referencing, Formulae in excel – average,
std. deviation, Charts, Renaming and Inserting worksheets, Hyper linking, Count function,
LOOKUP/VLOOKUP

**Task 3: Performance Analysis** - Features to be covered:- Split cells, freeze panes, group and
outline, Sorting, Boolean and logical operators, Conditional formatting
LaTeX and MS/equivalent (FOSS) tool Power Point

**Task 1:** Students will be working on basic power point utilities and tools which help them create basic power point presentation. Topic covered during this week includes: - PPT Orientation, Slide Layouts, Inserting Text, Word Art, Formatting Text, Bullets and Numbering, Auto Shapes, Lines and Arrows in both LaTeX and PowerPoint. Students will be given model power point presentation which needs to be replicated (exactly how it’s asked).

**Task 2:** Second week helps students in making their presentations interactive. Topic covered during this week includes: Hyperlinks, Inserting –Images, Clip Art, Audio, Video, Objects, Tables and Charts.

**Task 3:** Concentrating on the in and out of Microsoft power point and presentations in LaTeX. Helps them learn best practices in designing and preparing power point presentation. Topic covered during this week includes: - Master Layouts (slide, template, and notes), Types of views (basic, presentation, slide slotter, notes etc), and Inserting – Background, textures, Design Templates, Hidden slides.

**REFERENCE BOOKS:**
1. Comdex Information Technology course tool kit Vikas Gupta, *WILEY Dreamtech*
3. Introduction to Information Technology, ITL Education Solutions limited, *Pearson Education.*
4. PC Hardware - A Handbook – Kate J. Chase *PHI (Microsoft)*
5. LaTeX Companion – Leslie Lamport, *PHI/Pearson.*
CS312PC: PYTHON PROGRAMMING LAB

B.Tech. II Year I Sem.

L T P C
0 0 2 1

Prerequisites: A course on “Programming for Problem Solving”.

Course Objectives
- To be able to introduce core programming basics and program design with functions using Python programming language.
- To understand a range of Object-Oriented Programming, as well as in-depth data and information processing techniques.
- To understand the high-performance programs designed to strengthen the practical expertise.

Course Outcome
- Student should be able to understand the basic concepts scripting and the contributions of scripting language
- Ability to explore python especially the object-oriented concepts, and the built in objects of Python.
- Ability to create practical and contemporary applications such as TCP/IP network programming, Web applications, discrete event simulations

List of Experiments:
1. Write a program to demonstrate different number data types in Python.
2. Write a program to perform different Arithmetic Operations on numbers in Python.
3. Write a program to create, concatenate and print a string and accessing sub-string from a given string.
4. Write a python script to print the current date in the following format “Sun May 29 02:26:23 IST 2017"
5. Write a program to create, append, and remove lists in python.
6. Write a program to demonstrate working with tuples in python.
7. Write a program to demonstrate working with dictionaries in python.
8. Write a python program to find largest of three numbers.
9. Write a Python program to convert temperatures to and from Celsius, Fahrenheit. [ Formula : \( c/5 = (f-32)/9 \) ]
10. Write a Python program to construct the following pattern, using a nested for loop
    
        *
        **
        ***
        ****
        *****
        ****
        ***
        **
        *

11. Write a Python script that prints prime numbers less than 20.
12. Write a python program to find factorial of a number using Recursion.
13. Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether or not the triangle is a right triangle (Recall from the Pythagorean Theorem that in a right triangle, the square of one side equals the sum of the squares of the other two sides).
14. Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
15. Write a python program to define a module and import a specific function in that module to another program.
16. Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.
17. Write a program that inputs a text file. The program should print all of the unique words in the file in alphabetical order.
18. Write a Python class to convert an integer to a roman numeral.
19. Write a Python class to implement pow(x, n)
20. Write a Python class to reverse a string word by word.
MC309: GENDER SENSITIZATION LAB
(An Activity-based Course)

B.Tech. II Year I Sem.  

COURSE DESCRIPTION
This course offers an introduction to Gender Studies, an interdisciplinary field that asks critical questions about the meanings of sex and gender in society. The primary goal of this course is to familiarize students with key issues, questions and debates in Gender Studies, both historical and contemporary. It draws on multiple disciplines – such as literature, history, economics, psychology, sociology, philosophy, political science, anthropology and media studies – to examine cultural assumptions about sex, gender, and sexuality.

This course integrates analysis of current events through student presentations, aiming to increase awareness of contemporary and historical experiences of women, and of the multiple ways that sex and gender interact with race, class, caste, nationality and other social identities. This course also seeks to build an understanding and initiate and strengthen programmes combating gender-based violence and discrimination. The course also features several exercises and reflective activities designed to examine the concepts of gender, gender-based violence, sexuality, and rights. It will further explore the impact of gender-based violence on education, health and development.

Objectives of the Course:
- To develop students’ sensibility with regard to issues of gender in contemporary India.
- To provide a critical perspective on the socialization of men and women.
- To introduce students to information about some key biological aspects of genders.
- To expose the students to debates on the politics and economics of work.
- To help students reflect critically on gender violence.
- To expose students to more egalitarian interactions between men and women.

Learning Outcomes:
- Students will have developed a better understanding of important issues related to gender in contemporary India.
- Students will be sensitized to basic dimensions of the biological, sociological, psychological and legal aspects of gender. This will be achieved through discussion of materials derived from research, facts, everyday life, literature and film.
- Students will attain a finer grasp of how gender discrimination works in our society and how to counter it.
- Students will acquire insight into the gendered division of labour and its relation to politics and economics.
- Men and women students and professionals will be better equipped to work and live together as equals.
- Students will develop a sense of appreciation of women in all walks of life.
- Through providing accounts of studies and movements as well as the new laws that provide protection and relief to women, the textbook will empower students to understand and respond to gender violence.

UNIT-I: UNDERSTANDING GENDER
UNIT – II: GENDER ROLES AND RELATIONS
Two or Many? -Struggles with Discrimination-Gender Roles and Relations-Types of Gender Roles-Gender Roles and Relationships Matrix-Missing Women-Sex Selection and Its Consequences-Declining Sex Ratio. Demographic Consequences-Gender Spectrum: Beyond the Binary

UNIT – III: GENDER AND LABOUR

UNIT – IV: GENDER - BASED VIOLENCE
The Concept of Violence- Types of Gender-based Violence-Gender-based Violence from a Human Rights Perspective-Sexual Harassment: Say No! -Sexual Harassment, not Eve-teasing- Coping with Everyday Harassment- Further Reading: “Chupulu”.

UNIT – V: GENDER AND CULTURE
Gender and Film-Gender and Electronic Media-Gender and Advertisement-Gender and Popular Literature- Gender Development Issues-Gender Sensitive Language-Gender and Popular Literature - Just Relationships: Being Together as Equals

Note: Since it is Interdisciplinary Course, Resource Persons can be drawn from the fields of English Literature or Sociology or Political Science or any other qualified faculty who has expertise in this field from engineering departments.

- Classes will consist of a combination of activities: dialogue-based lectures, discussions, collaborative learning activities, group work and in-class assignments. Apart from the above prescribed book, Teachers can make use of any authentic materials related to the topics given in the syllabus on “Gender”.


ASSESSMENT AND GRADING:
- Discussion & Classroom Participation: 20%
- Project/Assignment: 30%
- End Term Exam: 50%
CS401PC: DISCRETE MATHEMATICS

B.Tech. II Year II Sem.  

Prerequisites: An understanding of Mathematics in general is sufficient.

Course Objectives

- Introduces the elementary discrete mathematics for computer science and engineering.
- Topics include formal logic notation, methods of proof, induction, sets, relations, graph theory, permutations and combinations, counting principles; recurrence relations and generating functions.

Course Outcomes:

- Ability to understand and construct precise mathematical proofs
- Ability to use logic and set theory to formulate precise statements
- Ability to analyze and solve counting problems on finite and discrete structures
- Ability to describe and manipulate sequences
- Ability to apply graph theory in solving computing problems

UNIT - I


UNIT - II

Basic Structures, Sets, Functions, Sequences, Sums, Matrices and Relations Sets, Functions, Sequences & Summations, Cardinality of Sets and Matrices Relations, Relations and Their Properties, n-ary Relations and Their Applications, Representing Relations, Closures of Relations, Equivalence Relations, Partial Orderings.

UNIT - III


Induction and Recursion: Mathematical Induction, Strong Induction and Well-Ordering, Recursive Definitions and Structural Induction, Recursive Algorithms, Program Correctness

UNIT - IV

Discrete Probability and Advanced Counting Techniques: An Introduction to Discrete Probability, Probability Theory, Bayes' Theorem, Expected Value and Variance

Advanced Counting Techniques: Recurrence Relations, Solving Linear Recurrence Relations, Divide-and-Conquer Algorithms and Recurrence Relations, Generating Functions, Inclusion-Exclusion, Applications of Inclusion-Exclusion

UNIT - V

Graphs: Graphs and Graph Models, Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Graph Coloring.

Trees: Introduction to Trees, Applications of Trees, Tree Traversal, Spanning Trees, Minimum Spanning Trees

TEXT BOOK:

REFERENCES BOOKS:
1. Discrete Mathematical Structures with Applications to Computer Science-J.P. Tremblay and R. Manohar, TMH,
SM402MS: BUSINESS ECONOMICS AND FINANCIAL ANALYSIS

B.Tech. II Year II Sem.  

Prerequisites: None

Course Objective: To learn the basic Business types, impact of the Economy on Business and Firms specifically. To analyze the Business from the Financial Perspective.

Course Outcome: The students will understand the various Forms of Business and the impact of economic variables on the Business. The Demand, Supply, Production, Cost, Market Structure, Pricing aspects are learnt. The Students can study the firm’s financial position by analysing the Financial Statements of a Company.

UNIT – I
Introduction to Business and Economics:

UNIT - II
Demand and Supply Analysis:
Elasticity of Demand: Elasticity, Types of Elasticity, Law of Demand, Measurement and Significance of Elasticity of Demand, Factors affecting Elasticity of Demand, Elasticity of Demand in decision making, Demand Forecasting: Characteristics of Good Demand Forecasting, Steps in Demand Forecasting, Methods of Demand Forecasting.

UNIT - III
Production, Cost, Market Structures & Pricing:
Production Analysis: Factors of Production, Production Function, Production Function with one variable input, two variable inputs, Returns to Scale, Different Types of Production Functions.
Cost analysis: Types of Costs, Short run and Long run Cost Functions.
Market Structures: Nature of Competition, Features of Perfect competition, Monopoly, Oligopoly, Monopolistic Competition.

UNIT - IV

UNIT - V
Financial Analysis through Ratios: Concept of Ratio Analysis, Liquidity Ratios, Turnover Ratios, Profitability Ratios, Proprietary Ratios, Solvency, Leverage Ratios (simple problems).
Introduction to Fund Flow and Cash Flow Analysis (simple problems).
TEXT BOOKS:

REFERENCE BOOKS:
CS403PC: OPERATING SYSTEMS

B.Tech. II Year II Sem. L T P C
3 0 0 3

Prerequisites:
- A course on “Computer Programming and Data Structures”.
- A course on “Computer Organization and Architecture”.

Course Objectives:
- Introduce operating system concepts (i.e., processes, threads, scheduling, synchronization, deadlocks, memory management, file and I/O subsystems and protection)
- Introduce the issues to be considered in the design and development of operating system
- Introduce basic Unix commands, system call interface for process management, interprocess communication and I/O in Unix

Course Outcomes:
- Will be able to control access to a computer and the files that may be shared
- Demonstrate the knowledge of the components of computer and their respective roles in computing.
- Ability to recognize and resolve user problems with standard operating environments.
- Gain practical knowledge of how programming languages, operating systems, and architectures interact and how to use each effectively.

UNIT - I

UNIT - II
System call interface for process management-fork, exit, wait, waitpid, exec

UNIT - III
Deadlocks - System Model, Deadlocks Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock
Process Management and Synchronization - The Critical Section Problem, Synchronization Hardware, Semaphores, and Classical Problems of Synchronization, Critical Regions, Monitors
Interprocess Communication Mechanisms: IPC between processes on a single computer system, IPC between processes on different systems, using pipes, FIFOs, message queues, shared memory.

UNIT - IV

UNIT - V
TEXT BOOKS:

REFERENCE BOOKS:
2. Operating System A Design Approach- Crowley, TMH.
4. UNIX programming environment, Kernighan and Pike, PHI/ Pearson Education
5. UNIX Internals -The New Frontiers, U. Vahalia, Pearson Education.
CS418PC: PRINCIPLES OF SOFTWARE ENGINEERING

B.Tech. II Year I Sem.  

Course Objectives

- The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects.
- Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and UML diagrams

Course Outcomes

- Ability to translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in a Software Requirements Document (SRD).
- Identify and apply appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices.
- Will have experience and/or awareness of testing problems and will be able to develop a simple testing report

UNIT - I
Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths.
A Generic view of process: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI), process patterns, process assessment, personal and team process models.
Process models: The waterfall model, incremental process models, evolutionary process models, the unified process.

UNIT - II
Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document.
Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.
System models: Context models, behavioral models, data models, object models, structured methods.

UNIT - III
Design Engineering: Design process and design quality, design concepts, the design model.
Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

UNIT - IV
Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging.
Product metrics: Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.

UNIT - V
Metrics for Process and Products: Software measurement, metrics for software quality.
Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan.
Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.

TEXT BOOKS:
3. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.

REFERENCE BOOKS:
CS412PC: OBJECT ORIENTED PROGRAMMING USING JAVA

B.Tech. II Year II Sem.  

Course Objectives:
- To introduce the object-oriented programming concepts.
- To understand object-oriented programming concepts, and apply them in solving problems.
- To introduce the principles of inheritance and polymorphism; and demonstrate how they relate to the design of abstract classes.
- To introduce the implementation of packages and interfaces.
- To introduce the concepts of exception handling and multithreading.
- To introduce the design of Graphical User Interface using applets and swing controls.

Course Outcomes:
- Able to solve real world problems using OOP techniques.
- Able to understand the use of abstract classes.
- Able to solve problems using java collection framework and I/o classes.
- Able to develop multithreaded applications with synchronization.
- Able to develop applets for web applications.
- Able to design GUI based applications.

UNIT - I
Object-Oriented Thinking- A way of viewing world – Agents and Communities, messages and methods, Responsibilities, Classes and Instances, Class Hierarchies- Inheritance, Method binding, Overriding and Exceptions, Summary of Object-Oriented concepts. Java buzzwords, An Overview of Java, Data types, Variables and Arrays, operators, expressions, control statements, Introducing classes, Methods and Classes, String handling.

UNIT - II
Packages- Defining a Package, CLASSPATH, Access protection, importing packages. Interfaces- defining an interface, implementing interfaces, Nested interfaces, applying interfaces, variables in interfaces and extending interfaces.

Stream based I/O (java.io)– The Stream classes-Byte streams and Character streams, Reading console Input and Writing Console Output, File class, Reading and writing Files, Random access file operations, The Console class, Serialization, Enumerations, auto boxing, generics.

UNIT - III
Exception handling - Fundamentals of exception handling, Exception types, Termination or resumptive models, Uncaught exceptions, using try and catch, multiple catch clauses, nested try statements, throw, throws and finally, built- in exceptions, creating own exception sub classes.

Multithreading- Differences between thread-based multitasking and process-based multitasking, Java thread model, creating threads, thread priorities, synchronizing threads, inter thread communication.

UNIT - IV
The Collections Framework (java.util)- Collections overview, Collection Interfaces, The Collection classes- Array List, Linked List, Hash Set, Tree Set, Priority Queue, Array Deque. Accessing a
Collection via an Iterator, Using an Iterator, The For-Each alternative, Map Interfaces and Classes, Comparators, Collection algorithms, Arrays, The Legacy Classes and Interfaces- Dictionary, Hashtable, Properties, Stack, Vector
More Utility classes, String Tokenizer, Bit Set, Date, Calendar, Random, Formatter, Scanner

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.


TEXT BOOKS:

REFERENCE BOOKS:
1. An Introduction to programming and OO design using Java, J. Nino and F.A. Hosch, John Wiley & sons
2. Introduction to Java programming, Y. Daniel Liang, Pearson Education.
CS406PC: OPERATING SYSTEMS LAB  
(Using UNIX/LINUX)

B.Tech. II Year II Sem.  

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Prerequisites:
- A course on “Programming for Problem Solving”.
- A course on “Computer Organization and Architecture”.

Co-requisite:
- A course on “Operating Systems”.

Course Objectives:
- To provide an understanding of the design aspects of operating system concepts through simulation
- Introduce basic Unix commands, system call interface for process management, interprocess communication and I/O in Unix

Course Outcomes:
- Simulate and implement operating system concepts such as scheduling, deadlock management, file management and memory management.
- Able to implement C programs using Unix system calls

List of Experiments:
1. Write C programs to simulate the following CPU Scheduling algorithms
   - a) FCFS  
   - b) SJF  
   - c) Round Robin  
   - d) priority
2. Write programs using the I/O system calls of UNIX/LINUX operating system (open, read, write, close, fcntl, seek, stat, opendir, readdir)
3. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance and Prevention.
4. Write a C program to implement the Producer – Consumer problem using semaphores using UNIX/LINUX system calls.
5. Write C programs to illustrate the following IPC mechanisms
   - a) Pipes  
   - b) FIFOs  
   - c) Message Queues  
   - d) Shared Memory
6. Write C programs to simulate the following memory management techniques
   - a) Paging  
   - b) Segmentation

TEXT BOOKS:

REFERENCE BOOKS:
2. Operating System - A Design Approach-Crowley, TMH.
4. UNIX Programming Environment, Kernighan and Pike, PHI/Pearson Education.
CS419PC: SOFTWARE ENGINEERING LAB

B.Tech. II Year II Sem. L T P C
0 0 3 1.5

Prerequisites
1. A course on “Programming for Problem Solving”

Co-requisite
1. A Course on “Software Engineering”

Course Objectives:
- To have hands on experience in developing a software project by using various software engineering principles and methods in each of the phases of software development.

Course Outcomes:
- Ability to translate end-user requirements into system and software requirements
- Ability to generate a high-level design of the system from the software requirements
- Will have experience and/or awareness of testing problems and will be able to develop a simple testing report

List of Experiments
Do the following 8 exercises for any two projects given in the list of sample projects or any other projects:
4. Study and usage of any Design phase CASE tool
5. Performing the Design by using any Design phase CASE tools.
6. Develop test cases for unit testing and integration testing
7. Develop test cases for various white box and black box testing techniques.

Sample Projects:
1. Passport automation System
2. Book Bank
3. Online Exam Registration
4. Stock Maintenance System
5. Online course reservation system
6. E-ticketing
7. Software Personnel Management System
8. Credit Card Processing
10. Recruitment system

TEXT BOOKS:
3. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.
CS408PC: JAVA PROGRAMMING LAB

B.Tech. II Year II Sem.         L   T   P   C
                                    0   0   2   1

Course Objectives:
- To write programs using abstract classes.
- To write programs for solving real world problems using java collection framework.
- To write multithreaded programs.
- To write GUI programs using swing controls in Java.
- To introduce java compiler and eclipse platform.
- To impart hands on experience with java programming.

Course Outcomes:
- Able to write programs for solving real world problems using java collection framework.
- Able to write programs using abstract classes.
- Able to write multithreaded programs.
- Able to write GUI programs using swing controls in Java.

Note:
1. Use LINUX and MySQL for the Lab Experiments. Though not mandatory, encourage the use of Eclipse platform.
2. The list suggests the minimum program set. Hence, the concerned staff is requested to add more problems to the list as needed.

List of Experiments:
1. Use Eclipse or Net bean platform and acquaint with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.

2. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, *, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.

3. a) Develop an applet in Java that displays a simple message.
b) Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named “Compute” is clicked.

4. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.

5. Write a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

6. Write a Java program for the following:
   Create a doubly linked list of elements.
   Delete a given element from the above list.
7. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with “Stop” or “Ready” or “Go” should appear above the buttons in selected color. Initially, there is no message shown.

8. Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.

9. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a Java program to display the table using Labels in Grid Layout.

10. Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired (Use Adapter classes).

11. Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables).

12. Write a Java program that correctly implements the producer – consumer problem using the concept of interthread communication.

13. Write a Java program to list all the files in a directory including the files present in all its subdirectories.

14. Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order

15. Write a Java program that implements Bubble sort algorithm for sorting in descending order and also shows the number of interchanges occurred for the given set of integers.

REFERENCE BOOKS
2. Thinking in Java, Bruce Eckel, Pearson Education.
The Constitution of India is the supreme law of India. Parliament of India cannot make any law which violates the Fundamental Rights enumerated under the Part III of the Constitution. The Parliament of India has been empowered to amend the Constitution under Article 368, however, it cannot use this power to change the “basic structure” of the constitution, which has been ruled and explained by the Supreme Court of India in its historical judgments. The Constitution of India reflects the idea of “Constitutionalism” – a modern and progressive concept historically developed by the thinkers of “liberalism” – an ideology which has been recognized as one of the most popular political ideology and result of historical struggles against arbitrary use of sovereign power by state. The historic revolutions in France, England, America and particularly European Renaissance and Reformation movement have resulted into progressive legal reforms in the form of “constitutionalism” in many countries. The Constitution of India was made by borrowing models and principles from many countries including United Kingdom and America.

The Constitution of India is not only a legal document but it also reflects social, political and economic perspectives of the Indian Society. It reflects India’s legacy of “diversity”. It has been said that Indian constitution reflects ideals of its freedom movement; however, few critics have argued that it does not truly incorporate our own ancient legal heritage and cultural values. No law can be “static” and therefore the Constitution of India has also been amended more than one hundred times. These amendments reflect political, social and economic developments since the year 1950. The Indian judiciary and particularly the Supreme Court of India has played an historic role as the guardian of people. It has been protecting not only basic ideals of the Constitution but also strengthened the same through progressive interpretations of the text of the Constitution. The judicial activism of the Supreme Court of India and its historic contributions has been recognized throughout the world and it gradually made it “as one of the strongest court in the world”.

Course content
1. Meaning of the constitution law and constitutionalism
2. Historical perspective of the Constitution of India
3. Salient features and characteristics of the Constitution of India
4. Scheme of the fundamental rights
5. The scheme of the Fundamental Duties and its legal status
6. The Directive Principles of State Policy – Its importance and implementation
7. Federal structure and distribution of legislative and financial powers between the Union and the States
8. Parliamentary Form of Government in India – The constitution powers and status of the President of India
9. Amendment of the Constitutional Powers and Procedure
10. The historical perspectives of the constitutional amendments in India
12. Local Self Government – Constitutional Scheme in India
13. Scheme of the Fundamental Right to Equality
14. Scheme of the Fundamental Right to certain Freedom under Article 19
15. Scope of the Right to Life and Personal Liberty under Article 21