

Jharkhand University of Technology

Ranchi, 834010

Diploma – Computer Science Engineering



SYLLABUS

**For Diploma Program in
Computer Science Engineering
(Effective from 2024-25)**

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

(3rd – SEMESTER)

Python Programming

Subject code – CSE301

1. Rationale

Computer programming is the core of the computer science and strong fundamentals of programming can give competitive edge in this technology driven world. It not only instils coding skills but also enhances problem solving ability. Python is one of the programming languages which is versatile and feature rich yet simple and easy to learn, has applications in various domains. Python programming sets the basis for further study of web development, data science, IoT, machine learning etc.

2. Course Outcomes: At the end of the Course, the student will be able to:

CO-01	Install the latest version of python distribution and configure it for an appropriate workspace as needed for a given project.
CO-02	Write a program by selecting python constructs needed to solve a given problem and then code, execute, test and debug the program to obtain the desired result.
CO-03	Demonstrate how a program can be optimized by using modular programming approach.
CO-04	Identify and resolve both syntactical and semantic errors in a given code snippet.

3. Course Content

Week	CO	PO	Lecture (Knowledge Criteria)	Tutorial (Activity Criteria)	Practice (Performance Criteria)
1	1,2	1,4	Fundamental Concepts: brief history; features; applications of python; python distributions; versions; python IDEs; Python interpreter; Execution of python programs, debugging python code; Indentation, Comments; best practices for python programming; Character set; tokens; keywords, variables, naming rules for variables, Assignment,	Refer Table 1	1. Setup python environment 2. Executing python: explore different ways to run python program 3. debug python code
2	2,4	1,2,4	Basics I/O operations Input- input (), raw_input() ; output – print (), formatting output. Datatypes		1. Code, execute and debug programs that a) Use i/o statements

			<p>Scalar type: Numeric (int, long, float, complex), Boolean, bytes, None; Type casting</p> <p>Operators Arithmetic, Comparison/Relational, Logical/Boolean, Bitwise; string operators; Expressions and operator precedence</p>	<p>b) Evaluate expressions and displays formatted output</p> <p>c) Evaluate expressions to examine the operator precedence</p> <p>2. Identify and resolve syntactic and semantic issues in the given code snippet</p>
3	2,4	1,2,4	<p>Control Flow: Conditional blocks If statement: general format; Multiway branching; Sufficient examples;</p>	<p>1. Identify and Code, execute and debug programs using conditional statements.</p> <p>2. Identify and resolve syntactic and semantic issues in the given code snippet</p>
4	2,4	1,2,4	<p>Control Flow: Loops While loop: general format; examples For loop: general format, examples. Range();nesting loops and conditional statements; Controlling loop execution: Break, continue, pass statements;</p>	<p>1. Code, execute and debug programs using loops.</p> <p>2. Code, execute and debug programs using loops and conditional statements</p> <p>3. Identify and resolve syntactic and semantic issues in the given code snippet</p>
5	2,4	1,2,4	<p>Data Collections Concept of mutability Set – features, declaration, initialization, operations, comprehension; Tuple-features; declaration, initialization, basic operations; indexing; slicing; built in functions; Nested tuples;</p>	<p>1. Code, execute and debug programs to perform following</p> <ul style="list-style-type: none"> ▪ set operations ▪ set comprehension <p>2. Code, execute and debug programs to perform following</p> <ul style="list-style-type: none"> ▪ basic operations on tuples ▪ tuple indexing and slicing <p>3. Identify and resolve syntactic and semantic issues in the given code snippet</p>
6	2,4	1,2,4	<p>List features; declaration, initialization, basic operations; indexing; List iterations; Slicing; built in functions; Nested Lists; Comprehensions; Applications</p>	<p>1. Write code snippet to perform following on List</p> <ul style="list-style-type: none"> ▪ basic operations on List ▪ indexing and slicing ▪ comprehension

					2. Identify and resolve syntactic and semantic issues in the given code snippet
7	2,4	1,2,4	Dictionary features; declaration, initialization, basic operations; indexing; adding and removing keys, iterating through dictionaries; built in functions; Comprehensions; Applications		1. Code, execute and debug programs to perform basic operations on Dictionary 2. Code, execute and debug programs to perform Dictionary indexing Iterating comprehension 3. Identify and resolve syntactic and semantic issues in the given code snippet
8	2,4	1,2,4	Arrays and Strings Arrays: features; create, initialize, indexing, traversal, manipulation; Strings: create, assign, indexing, built in functions;		1. Code, execute and debug programs to perform string manipulation 2. Code, execute and debug programs to perform array manipulation 3. Identify and resolve syntactic and semantic issues in the given code snippet
9	2,3,4	1,2,4	Functions Need of function; types; define function, calling function, function arguments; return and yield; None keyword; Scope of variables; Recursion; anonymous functions; sufficient examples;		1. Code, execute and debug programs to solve the given problem using built in functions 2. Code, execute and debug programs to solve the given problem by defining a function 3. Code, execute and debug programs to solve the given problem using recursion 4. Define anonymous function and code to solve the given problem 5. Identify and resolve syntactic and semantic issues in the given code snippet
10	2,3,4	1,2,4	Modules and Packages		1. Create Modules and Packages

			Why modules? Module creation; Importing modules; Module Namespace; Packages: basics; path setting; Package_init_.py Files; Commonly used modules: Math, random; Emoji;		2. Code, execute and debug programs using built in modules
11	2,3,4	1,2,4	NumPy Brief about NumPy module; NumPy arithmetic functions; NumPy array manipulation functions; NumPy statistical functions; Pandas Introduction, series, data frame; Create dataframes; formatting data; fundamental data frame operations;		1. Code, execute and debug programs using NumPy module. 2. Code, execute and debug programs using series. 3. Code, execute and debug programs using dataframes. 4. Identify and resolve syntactic and semantic issues in the given code snippet
12	2,3,4	1,2,4	Files Concept; features; file operations; Opening Files; Closing Files; Writing to Files; Reading to Files; File methods; Working with files using data frame.		1. write code snippet to perform following operations on different types of files <ul style="list-style-type: none"> read file write to file. 2. Write code to perform file operations using dataframes on different file types. 3. Identify and resolve syntactic and semantic issues in the given code snippet
13	2,3,4	1,2,4	Error and Exception Handling: Python errors; exceptions: built in, user defined. How to catch exceptions? Raising exceptions;		1. Integrate exception handling into above code 2. Write code snippet to raise exceptions 3. Identify and resolve syntactic and semantic issues in the given code snippet
Total in hours			39	13	52

***PO = Program outcome as listed and defined in year 1 curriculum**

Table 1: Suggestive activities for tutorials (the list is only shared as an example and not inclusive of all possible activities for that course. Student and faculty are encouraged to choose activities that are relevant to the topic and the availability of such resources at their institution)

Sl no	Activity
1	1. Compare and contrast excel and python 2. Identify various python IDEs and identify differences between them.
2	1. Identify use cases like reading student name and contact details and display in a required format 2. Compare and contrast input () and raw_input() and identify its appropriate use.

	3. Identify use of operators and develop algorithms to solve the same 4. Compare and contrast different types of operators
3	1. Identify use cases that involve decision making and develop algorithms to solve the same 1. Identify common syntactical errors when using control flow statements
4	1. Identify use cases that involve iteration and develop algorithms to solve the same 2. Compare and contrast different types of loops 2. Identify common syntactical errors when using loops
5	1. Identify use cases and solve them using sets 2. Identify use cases and solve them using tuples 3. Identify common syntactical errors when working with sets and tuples
6	1. Identify use cases and solve them using List 2. Identify common syntactical errors when working with List 3. Reimplement built in list functions
7	1. Identify use cases and solve them using dictionary 2. Reimplement built in dictionary functions 3. Identify common syntactical errors when working with dictionary
8	1. Identify use cases and solve them using arrays 2. Reimplement built in string functions 3. Identify common syntactical errors when working with arrays and strings
9	1. Optimize previously written programs by using modular programming approach
10	1. Identify and present pros and cons of modules and packages 1. Explore and present python built in modules.
11	1. Identify the applications of Pandas 2. Perform data analysis using Pandas module on a dataset such as .
12	1. Identify use cases on files concept and develop algorithms to solve the same 2. Explore regular expressions and present how they can be used for file manipulation
13	3. Compare and contrast error and exception. 4. Rewrite the programs using exceptions if needed

4. Reference:

Sl. No.	Description
1	Core python programming, Wesley J. Chun Publisher: Prentice Hall PTR
2	Fluent Python by Luciano Ramalho
3	https://www.softcover.io/read/e4cd0fd9/conversational-python
4	https://realpython.com/
5	https://www.python-course.eu/
6	https://www.datacamp.com/
7	https://www.w3schools.com/

Computer Hardware, Maintenance and Administration

Subject code – CSE302

1. Rationale

Professional computer maintenance ensures computer hardware and software systems run efficiently to increase productivity while lowering the chances of downtime. This course aims to help understand the internal working of computers/laptops and prepare the student for a role as an entry-level IT support technician. This course sets a basis for different facets of information technology like computer hardware, software, trouble shooting and customer service.

2. Course Outcomes: At the end of this course, students will be able to:

CO-01	Assemble a computer as per given technical specifications following all necessary safety protocols and install, configure and setup an administrator for a Windows Operating System.
CO-02	Diagnose a computer using the right diagnostic tools, identify the hardware problem and troubleshoot to resolve the problem following all safety protocols.
CO-03	Diagnose an installed software using the right diagnostic tools, identify the bug/issue, troubleshoot to resolve bugs/issues and ensure all data and applications are backed up before troubleshooting.
CO-04	Explain e-waste protocols to be followed while disposing computer hardware, to ensure compliance with all required state pollution control board regulations.

3. Course Content

Week	CO	PO	Lecture (Knowledge Criteria)	Tutorial (Activity Criteria)	Practice (Performance Criteria)
1	01	1, 4	I/O devices and Interfaces Types of I/O devices and ports on a standard PC for connecting I/O devices. Function of serial port, parallel port, and brief principle of communication through these ports, types of devices that can be connected and interface standards. Explain basic cable types, features and their purposes. Importance of USB and HDMI interfaces- Types and Features. Working of Common	Refer Table 1	1. Hardware Identification: <ul style="list-style-type: none">• Computer Case- Types, Features- Front panel, back panel; A look inside the computer case.• Identify the front and rear panel controls and ports on a PC cabinet.• Identify and understand different cables and connectors: Video cables- VGA, HDMI, Mini-HDMI, Display port, DVI; Peripheral cables- Serial; Hard drive cables- SATA, PATA, IDE,

			Input/Output devices- Keyboard, Mouse, display monitor, printer and speaker.		SCSI; Adapters-DVI to HDMI, USB to ethernet, DVI to VGA. • Installation of a local printer. • Shared printer. • Installing wireless and cloud printers.
2	01	1,4	Power supplies Safety Basics: safety protocols; anti-static basic hand tools, Know the danger of static electricity, power variation; Precautions to be taken while mounting and unmounting power supply into/ from the cabinet; DC power source to PC - Need for SMPS, Specifications, Rating of SMPS based on type of motherboard and devices used (AT/ATX, Micro ATX, mini ATX, higher wattage PSU for gaming PC), color coding adopted, Types of connectors used- ATX, ATX12V, Molex, SATA, PCIe; Output voltage levels, measuring technique; choosing power supply based on wattage rating; Heat sink; 80 plus rating system; Modular power supply. Symptoms of SMPS failure; Common problems from a faulty SMPS. Troubleshooting Power supplies.		1. Unmount the power supply from PC cabinet. Identify the types of output connectors. 2. Identify output voltages using color coding. 3. Measure voltage levels using multi meter. 4. Mount the power supply into the PC cabinet, connect different components and test PC. 5. Troubleshoot Power supply through SMPS fan. 6. Diagnose power supply faults using PSU Tester.
3	1	1,4	Basic Electronics –Electricity, Electrical quantities- Voltage, current and resistance; Active components: Diodes- PN junction diodes, biasing conditions- forward bias and reverse bias, transistors- BJT, MOSFET; Passive components-Resistors, capacitors, inductors, transformers, sensors, and transducers; Integrated Circuits Digital Electronics- PWM, opto-coupler; checking AC to DC converter.		1. Identify the electrical and electronic components used in a computer and tabulate them as active and passive components. 2. Identify the working and non-working state of basic components and semiconductor devices. 3. Using multi meter- Check Output voltage of basic components and semiconductor devices. 4. Check different voltage levels of opto-coupler, PWM and rectifier.
4	01,03	01,04	Mother board – Example Motherboards; Functional description of mother board; specification and variation. Form factor- what is Motherboard Form factor? Types and features of Motherboard form factors- ATX, Micro-ATX, Mini-ITX,		1. Precautions to be taken before removing the mother board from PC cabinet. 2. Using the CPUID CPU-Z tool, find different features of CPU. 3. Check the Electric flow path and data flow path 4. Windows resource monitor

			Nano-ITX, and Pico-ITX. Functional components of Motherboard; CPU and CPU socket-Types of sockets; Overview of microarchitecture of INTEL and AMD CPU.		5. Using the CPUID CPU-Z tool, identify the CPU cache features of your working system.
5	01, 03	01, 04	Chipsets- Function, Types and Features. Buses- System bus architecture Importance of POST; UEFI – why is it required, possible configurations through UEFI. IDE ports; Methods of adding SCSI drives. CMOS battery: Why? Its specifications. Impact of removing the battery from mother board.		<ol style="list-style-type: none"> 1. Identify system faults using POST diagnostics card. 2. Understand basic onboard configuration through UEFI. 3. Test different motherboards to determine support for UEFI 4. Replace the CMOS battery in a computer following the procedures. 5. Understand and modify BIOS settings and observe the consequences of CMOS failure.
6	01, 03	01, 04	Memory– Memory Units (B, KB, MB, GB, TB), memory locations and address space, Access methods, Memory Classification. Main memory Types and Features. Auxiliary memory – Types and features. Memory modules.		<ol style="list-style-type: none"> 1. Identify RAM chips and HDD/SSD, study their features and note their technical specifications. 2. Identify SIMM and DIMM memory modules, their number of pins, specs and type. 3. Identify the interface type of a hard drive and connect it to a PC for data recovery.
7	01, 03	01, 04	RAM Technology- SDRAM, DDR, DDR2, DDR3, DDR4 – Clock speed, Bandwidth, Memory speed rating, PC speed rating; RAM capacity- single- sided and double-sided RAM, Channels; RAM features- Parity/ECC RAM, SODIMM, SPD chip. Mass storage media- Hard drive, Principle of working, reliability, performance, SSD, optical drive; Logical Block Addressing (LBA); Memory capacity- physical and logical addressing; M.2 drives, SATA, NVMe. Causes of Hard drive failure; Signs of failure; Backup and recovery of data;		<ol style="list-style-type: none"> 1. Use CPUID-CPUZ tool to identify capacity, speed, technology, and related features of RAM. 2. Check for RAM and Motherboard compatibility and install additional RAM stick. 3. Find on Windows system properties to check the RAM for correct installation. 4. Query the SPD RAM chip to identify all 5. possible information using CPUID CPUZ.
8	02, 03		Windows Installation – Overview windows 10; general features; Versions; architecture; prerequisites for windows 10 installation: hardware compatibility, BIOS compatibility, driver requirements. Clean install of		<ol style="list-style-type: none"> 1. Windows Installation Inspect prerequisites for windows 10 installation on a given computer. 2. Perform clean installation. 3. Upgrade to windows 10. 4. Create dual boot for a given system, learn and rectify errors

			windows 10; upgrade to windows 10; disk partitioning; troubleshooting installation problems; Multiple boot options; windows service packs. Imaging: create a Windows system image; How to Backup/Restore your Windows partition with the bootable image.		in dual boot. 5. Practice on recovery partition. 6. Practice 10 registry tweaks. 7. Practice disk management utilities.
9	01, 02, 03		File system overview, types, properties, conversion from one filesystem to another, configuration. Configuring system and data recovery: Recover files, recover apps and the registry; recover windows 10. Configure and manage windows updates: auto/manually; testing and troubleshooting updates. Monitor and manage: Performance monitoring; optimize windows services; tune scheduled tasks. Customizing windows desktop.		1. Practice data recovery methods 2. Working with task manager to troubleshoot configuration and other performance related issues. 3. Working with task scheduler. 4. Customizing windows desktop.
10	01, 03	01, 04, 07	Windows Command line; PowerShell; basic commands; writing simple PowerShell scripts. File security.		1. Execute basic commands in Windows using command prompt and PowerShell like listing the drives in a system, creating a new file, removing a file or directory, retrieving the list of processes and services, etc., 2. Use command line to encrypt and decrypt files and folders.
11	02, 03, 04	01, 04	Portable computing- Troubleshooting Laptops- I Difference between laptop and desktop Motherboards; Checking Power connector and adaptor pins, AC adapter of Laptop circuit diagram, Fault finding; troubleshooting voltage transfer section, AC-DC conversion, Generation of stand by voltage, fault finding- No power ON, power switch, Battery charging circuit;		1. Observe the layout of a laptop and compare it with a desktop. 2. Follow/review manufacturer maintenance guide for repair and maintenance. 3. Power Issues: Battery not charging, No power. 4. Trouble shoot computer hardware issues in the following scenarios- -Unexpected shutdowns. -Lockups -POST & Boot -Continuous reboot -No Power -Loud Noises. -Intermittent device failure -Smoke and burning smell 5. -Indicator lights.

12	03, 04,	01, 04	Troubleshooting Laptops- II Dual MOSFET pin details, Two N- channel MOSFET in place of dual MOSFET, one p-channel and one N- channel MOSFET, problems and faultfinding; CPU voltage generation circuits, keyboard interface, problem and fault finding. Touch pad connector, BIOS details, SATA HDD details, Audio section, internal display, LED screen pin details. Malware mitigation: introduction, types, symptoms, malware removal;		1. Perform the same operations in week#7 on a laptop. 2. LCD display trouble-shoot: No display/dim video/flickering video. 3. Wireless troubleshooting: Multiple antennae, check presence of wireless cards. 4. Scan and remove malwares in each computer or mobile device. 5. Perform Antivirus and anti-malware updates.
13	04,	01, 04	E-waste management: What is EEE and E-waste? Different scenarios of E-waste management, StEP initiatives to solve e-waste problems, impact of e-waste on health of children and workers. E-waste management in India: EPR and e-waste, the informal sector in e-waste management, Technologies for e-waste management, Financing e-waste management systems- Key steps, milestones to achieve a robust E-waste system. Case studies.		<ul style="list-style-type: none"> Visit https://greene.gov.in and https://kspcb.karnataka.gov.in/to find the latest regulations and policies taken up by the Government of India. Visit a nearby e-waste management plant and understand the management process.
Total in hours			39	13	52

*PO = Program outcome as listed and defined in year 1 curriculum

Table 1: Suggestive activities for tutorials (the list is only shared as an example and not inclusive of all possible activities for that course. Student and faculty are encouraged to choose activities that are relevant to the topic and the availability of such resources at their institution)

SL NO	Activity
1	1. Study multipurpose cables used with different models of computers and their applications. 2. Identify the cables and list the devices that can be connected to computer using these cables. 3. Identify commercially available brands of keyboard, mouse and monitor and their distinguishable features.
2	1. Identify preventive measures that help to eliminate or reduce electrostatic discharge. 2. Describe effective ways to reduce the risk of injury or damage while working with respect to power supply in computer systems. 3. Make your observations on why a technician may choose to use a UPS instead of a surge suppress or to protect a computer. 4. Identify the components that are powered by the PSU. 5. Identify the common causes of PSU/SMPS failure and explain how it can be maintained in a healthy state.

	6. Explain the factors on which the PSU wattage is decided. 7. Describe the features of different Power supplies available commercially that support a gaming PC.
3	1. Calculate the value of resistors onboard using the color code. 2. Identify the terminals of a BJT and MOSFET.
4	1. Identify the units MHz and GHz with respect to CPU. 2. Compare and contrast the characteristics of different motherboard formfactors. 3. Identify and present the factors to select an appropriate Motherboard based on the purpose-Basic applications, Gaming, Workstations/servers. 4. Study the features and specifications of the processors (Intel: Pentium family, dualcore, quad core, core 2 duo, i3, i5, i7, i9 and AMD processors). 5. Study the CPU benchmarks of the trending processors. 6. Compare features of CPU and GPU.
5	1. Explain bus standards with respect to evolution, speed, and recent trends (ISA, PCI, AGP). 2. Explain how to clear CMOS password. 3. Explain the importance of UEFI and configuration settings for – a Overclocking of CPU frequencies b Set RAM timings 4. Setting BIOS passwords Specify boot options.
6	1. Describe the classification of memories. 2. Study different types of memory devices and features that are commercially available.
7	1. Study the characteristics of different types of SSDs. 2. Compare DDR4 and DDR5 memory. 3. Study and identify what happens when the hard drive fails.
8	1. Identify system requirements for Windows 10 installation. 2. Study and list the features of different versions of windows 10. 3. Explore file backup in Windows file history feature.
9	1. Study best practices followed in scheduled backups and scheduled disk maintenance. 2. Study the necessity of TPM in Windows 11.
10	1. Compare relative merits and demerits of the two command shells.
11	1. Relate the components of a laptop to a desktop and compare their features. 2. Identify the upgradable and non- upgradable components in laptops.
12	1. Document the common faults that may occur on the motherboard.
13	1. Study the e-waste rules 2016 and their amendments there-off as prescribed by the Karnataka state pollution control board and explain the process for the polytechnic to follow safe disposal of e-waste. 2. Study the global e-waste key statistics. 3. Sketch the e-waste policies and regulations, you think, must be regulated at each stakeholder's level.

4. Reference:

Sl. No.	Description
1	https://www.dell.com/support/kbdoc/en-in/000139662/what-does-the-msconfig-utility-do-in-windows-7-on-your-dell-pc#TOC
2	https://www.google.co.in/amp/s/www.ufsexplorer.com/amp/articles/how-to/connect-sata-disks-instruction.php
3	http://www.qiguaninc.com/met/faq/faq35_en.html
4	https://www.ciscopress.com/articles/article.asp?p=2999386&seqNum=3
5	https://www.crucial.in/articles/pc-builders/what-is-computer-hardware
6	https://www.udemy.com
7	https://www.pluralsight.com

Computer Networks

Subject code – CSE303

1. Rationale

The computer networking skills are essential in today's information and communication technology driven world. It enables students with essential skills and knowledge to explore the world of communication and networking for further study and career.

2. Course Outcomes: At the end of the course, the student will be able to:

CO-01	Select an appropriate communication technology for a given network and ensure optimal performance by addressing issues arising from transmission impairments.
CO-02	Design a network for a given specification by using the right network components, devices, topologies, protocols and software.
CO-03	Design, build, test and troubleshoot a SOHO network for a given premises.
CO-04	Demonstrate the configuration of IP address, routing, subnetting, client-server interaction (TCP, UDP) and DNS for a given WAN network using a network simulator and troubleshoot common network issues

3. Course Content

Week	CO	PO	Lecture (Knowledge Criteria)	Tutorial (Activity Criteria)	Practice (Performance Criteria)
1	01	1, 2	Electromagnetic waves - Generation of electromagnetic waves and their properties Electromagnetic spectrum - classification and its applications Communication Systems - Basic elements of communication systems with block diagram, List commonly used terms in electronic communication systems, Data representation, Data flow, Modulation, Demodulation Analog and Digital Signals, Periodic and Non-Periodic Signals, Sine Wave, Phase, Wavelength, Digital Signals, Bit-rate, Bit-length.	Refer Table 1	1. Build a circuit to Generate and detect of BASK signal and BFSK signal using communication kit.
2	01	1,2	Transmission Impairment – Attenuation, Distortion and Noise,		1. Explore all ISP in your area/locality and select best internet ISP/plan

			<p>Performance - Bandwidth, Throughput, Latency, Jitter (Basic concepts only).</p> <p>Transmission Modes – Parallel and Serial Transmission. Asynchronous and Synchronous Transmission.</p> <p>Satellite communication- Introduction, advantages and disadvantages</p>		<p>based on cost and performance.</p> <p>2. Test the download/upload speed in your computer/mobile phone also check type, bandwidth and ISP.</p> <p>3. Explore Bluetooth, Wifi, NFC in your smartphone and note their key technical attributes (Radio spectrum band, range, pathloss, throughput, mode etc)</p>
3	01, 02	1, 2, 3	<p>Perspectives on Networking – End user perspectives on Networking and Internet, Overview of Networking. Categories of networks - LAN, MAN, WAN, Internetworking (Illustrate Network from LAN connected using a HUB to Internetwork).</p> <p>The communication rules (Method, language, Confirmation) – Protocols, the Internet.</p> <p>The Network Standard Organizations, Protocol Stack.</p> <p>OSI Model : OSI Layers and Their Functions, OSI Layering Concepts and Benefits, OSI Encapsulation Terminology.</p>		<p>1. My Protocol Rules Objectives</p> <p>a) Relate computer network protocols to the rules that you use every day for various forms of communication.</p> <p>b) Define the rules that govern how to communicate in a group of students.</p> <p>c) Play the communication game.</p> <p>d) List what would happen if the sender and receiver did not agree on the details of the protocol.</p> <p>2. Manual and Automatic address assignment (Windows)</p> <p>a) IPv4 address</p> <p>b) Subnet mask</p> <p>c) DNS</p> <p>3. Manual and Automatic address assignment (Android)</p> <p>a) IPv4 address</p> <p>b) Subnet mask</p> <p>c) DNS</p>
4	01, 02	1, 2, 3	<p>TCP/IP Networking Model - History Leading to TCP/IP, Overview of the TCP/IP Networking Model.</p> <p>TCP/IP Application Layer, HTTP Overview, HTTP Protocol Mechanisms.</p> <p>TCP/IP Transport Layer, TCP Error Recovery Basics, Same-Layer and Adjacent-Layer Interactions.</p> <p>TCP/IP Network Layer, Internet Protocol and the Postal Service, Internet Protocol Addressing Basics, IP Routing Basics.</p> <p>TCP/IP Link Layer (Data Link Plus Physical), TCP/IP Model and Terminology, Data</p>		<p>1. Organize and play games to understand working of TCP/IP like: Create 2 group of students, each playing role of a layers of TCP/IP (intermediate network devices roles can also be considered). Start the communication between two with a sender and receiver.</p> <p>2. Determine the IP Address Configuration of a</p>

			Encapsulation Terminology. Names of TCP/IP Messages.		Computer (Windows) and Test the Network Interface TCP/IP Stack (Ping).
5	02, 03	1, 2, 4,	Hardware and Software components of Network - Common network devices - Computers, Access points, Hub, Switch, Router, repeaters, NIC, Modem. LAN Cables – Co-axial, twisted pair, optical fibre, LAN connectors- co-axial cable, and twisted pair cable, optical fibre, Connectors, Firewall, Firm wares, ISPs.		1. Demonstrate working of common network devices. 2. Demonstrate different network cables and connectors. 3. Install and configure NIC. 4. Crimping of RJ45: Straight and Cross. a) Punching Cat 6 cable to I/O Box. Use punching tool. b) Check connectivity using LAN tester
6	01, 02	1, 2, 3, 4,	Overview of network topologies - Basic topologies- bus, ring, star, mesh and hybrid. Network Simulator: Network simulator like Packet Tracer, Installation, User Interface. Deploy devices and cables GUI and CLI Configuration. Configure end Devices		1. Install Network simulator like Cisco packet tracer. 2. Create simple network in simulator. 3. Create and Demonstrate all possible network topologies using simulator.
7	02, 03	1, 2, 3, 4,	An Overview of LANs - Typical SOHOLANs, Typical Enterprise LANs, The Variety of Ethernet Physical Layer Standards, Consistent Behaviour over All Links Using the Ethernet Data Link Layer. Building Physical Ethernet Networks with UTP - Transmitting Data Using Twisted Pairs, Breaking Down a UTP Ethernet Link, UTP Cabling Pinouts for 10BASE-T and 100BASE-T, Straight-Through Cable Pinout, Crossover Cable Pinout, Choosing the Right Cable Pinouts, UTP Cabling Pinouts for 1000BASE-T, Sending Data in Ethernet Networks.		1. Build a physical Ethernet LAN Network and demonstrate file sharing, printer sharing. 2. Install and configure wireless access point over the LAN. 3. Use pathping command to find actual path between source to destination with information about network latency/delay & network loss.
8	02, 03	1, 2, 3, 4,	Ethernet Data Link Protocols – The Rise of Ethernet, The Ethernet MAC address and Ethernet Addressing, Identifying Network Layer Protocols with the Ethernet Type Field, Error Detection with FCS. Encapsulation, Ethernet Frame. Hierarchical Network Design – Physical and logical addresses, Benefits of a Hierarchical Design, Access, Distribution, and Core layers Sending Ethernet Frames with Switches and Hubs, Sending in Modern Ethernet LANs Using Full-Duplex, Using Half-Duplex with LAN Hubs. Ethernet access layer devices – Hub, Switch, The MAC address table,		1. Determine the MAC Address of a Host(PC and Phone). 2. View Wireless and Wired NIC Information and make a table explaining each. 3. Configure and install a ethernet switch/Hub (Use simulator if hardware devices are not available) 4. Create/model a simple Ethernet network using 3 hosts and a switch, Observe traffic behavior on the network and Observer

			Ethernet Broadcast and Broadcast domain, ARP.		data flow of ARP broadcasts and pings.
9	02, 03, 04	1, 2, 3, 4,	<p>Routing: The Need for Routing - Criteria for Dividing the Local Network - Now We Need Routing</p> <p>Overview of Network Layer Functions - Network Layer Routing (Forwarding) Logic, Host Forwarding Logic: Send the Packet to the Default Router, Routing Data Across the Network, Delivering Data to the End Destination, How Network Layer Routing Uses LANs and WANs, IP Addressing and How Addressing Helps IP Routing, Routing Protocols.</p> <p>IPv4 Addressing - Rules for IP Addresses, Rules for Grouping IP Addresses, Class A, B, and C IP Networks, The Actual Class A, B, and C IP Networks, IP Subnetting, How to create subnets, Subnet mask, CIDR, variable length subnet mask.</p>		<p>1. Build a simple peer-to-peer network and verify physical connectivity and Assign various IPv4 addresses to hosts and observe the effects on network communication</p> <p>2. Configure IP addresses of a network (real or simulated) and ping across to test and troubleshoot.</p> <p>3. Subnetting of a network (either using real network or in Simulator).</p> <p>4. Connect to web server using simulator, Observe how packets are sent across the Internet using IP addresses.</p>
10	02, 03, 04	1, 2, 3, 4, 7	<p>IPv4 Routing - IPv4 Host Routing, Router, Forwarding Decisions and the IP Routing Table, The default gateway, A Summary of Router Forwarding Logic, A Detailed Routing Example.</p> <p>Routing Protocols - IPv4 Routing Protocols – static and dynamic.</p> <p>Other Network Layer Features - Using Names and the Domain Name System, The Address Resolution Protocol, ICMP Echo and the ping Command.</p> <p>DHCP – Static address assignment, Dynamic address assignment, DHCP servers.</p>		<p>1. Implement simple static routing.</p> <p>2. Troubleshooting of IP Addressing-</p> <p>a) Change a routing table entry</p> <p>b) Wrong address</p> <p>c) incorrect subnet mask</p> <p>3. Configure and test DHCP on a wireless router (real or simulated)</p>
11	02, 03, 04	1, 2, 3, 4,	<p>Pv4 and IPv6 Address Management - Network Boundaries -Gateways to Other Networks, Routers as Gateways.</p> <p>Network Address Translation - Introduction, NAT operation.</p> <p>IPv4 Issues - Need of IPv6, Ipv4 vs IPv6, IPv4 and IPv6 Coexistence.</p> <p>IPv6 features - IPv6 Address Representation.</p> <p>Concept of Virtual LAN's(VLAN's)</p>		<p>1. Packet Tracer - Examine NAT on a Wireless Router</p> <p>2. Identify IPv6 Addresses</p> <p>a) Identify the different types of IPv6 addresses.</p> <p>b) Examine a host IPv6 network interface and address.</p> <p>c) Practice IPv6 address abbreviation.</p> <p>3. Setup, configure and test VPN in your smartphone.</p>
12	02, 03, 04	1, 2, 3, 4,	<p>The Client Server relationship – Client Server interaction. URI, URN, URL</p> <p>TCP/IP Layer 4 Protocols: TCP and UDP - Transmission Control Protocol, Multiplexing Using TCP Port Numbers, Popular TCP/IP Applications, Connection Establishment and Termination, User Datagram Protocol.</p>		<p>1. Create a client – server model in simulator and observe the client interaction between the server and PC using packet tracer.</p> <p>2. Observe DNS Name Resolution</p>

			Port Numbers – TCP and UDP. Socket pairs, the netstat command. Application Layer Services - Common Network Application Services, Domain Name System, Domain Name Translation, DNS Servers, HTTP, Web Clients and Servers, FTP, Virtual Terminals, Remote Access with Telnet or SSH, Telnet, Security Issues with Telnet, SSH, Email- Email protocols, Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP3), Internet Message Access Protocol (IMAP4).		a) Observe the conversion of a URL to an IP address. b) Observe DNS lookup using the nslookup command. 3. Use simulator to demonstrate Telnet and SSH
13	02, 03, 04	1, 2, 3, 4,	Troubleshoot Common Network Problems - The Troubleshooting Process, Network Troubleshooting Overview, Gather Information - Nature of problem, Equipment, Configuration and Topology, Previous Troubleshooting. Structured Troubleshooting Methods - Bottom-Up, Top-Down, Divide-and-Conquer, Follow-the-Path, Substitution, Comparison, Educated Guess. Guidelines for Selecting a Troubleshooting Method Troubleshoot Wireless Issues - Causes of Wireless Issues, Authentication and Association Errors.		1. Demonstrate troubleshooting Commands with a scenario- ipconfig, ping , netstat, tracert, nslookup. 2. Interpret the output of commonly used network command line utilities and Determine which network utility can provide the necessary information to perform troubleshooting activities in a bottom-up troubleshooting strategy 3. Physical Layer Problems - Common Layer 1 Problems, how to use the five senses to troubleshoot, Wireless Router LEDs, Cabling Problems 4. Common Internet Connectivity Issues - DHCP Server Configuration Errors, Check Internet Configuration, Check Firewall Settings.
Total in hours			39	13	52

*PO = Program outcome as listed and defined in year 1 curriculum

Table 1: Suggestive activities for tutorials (the list is only shared as an example and not inclusive of all possible activities for that course. Student and faculty are encouraged to choose activities that are relevant to the topic and the availability of such resources at their institution)

Sl. No	Activity
1	Prepare a report on advanced communication systems and suggest best way to connect remote villages of India.
2	Prepare a report on Communication satellites of Indian Space Research Organization.
3	Prepare a presentation on 5 networking protocols being used in your smart phone.
4	My Protocol Rules Objectives a) Relate computer network protocols to the rules that you use every day for various forms of communication. b) Define the rules that govern how you send and interpret text messages.

	c) Explain what would happen if the sender and receiver did not agree on the details of the protocol.
5	My Local Network a) Record all the different network-attached devices in your home or classroom. b) Investigate how each device connects to the network to send and receive information. c) Create a diagram showing the topology of your network. d) Label each device with its function within the network.
6	Detailed study of Packet tracer and present the same.
7	Trace a Route a) Determine network connectivity to a destination host b) Trace a route to a remote server using tracert
8	Presentation on wireless Ethernet protocols
9	1. Calculate whether destination address is local or remote using IP address. 2. Calculate whether destination address is local or remote using mask. 3. Use logical AND to determine network address
10	Identify IPv6 Addresses a) Identify the different types of IPv6 addresses. b) Examine a host IPv6 network interface and address. c) Practice IPv6 address abbreviation.
11	Prepare a report on ICANN List the popular port numbers with their use.
12	Prepare a report on popular application layer protocols and present the same.
13	Identify and correct any misconfiguration of a wireless device (Scenario : A small business owner learns that a wireless user is unable to access the network. All the PCs are configured with static IP addressing. Identify and resolve the issue)

4. Reference:

Sl. No.	Description
1	www.netcad.com
2	Computer Networks - Behrouz A. Forouzan
3	www.howtonetwork.com
4	vlab.co.in

Database System Concepts and PL/SQL

Subject code – CSE304

1. Rationale

Data, factual information, is the main driving force that is changing the face of our world. Database is an organized collection of related data which is stored and accessed electronically using a computer. Database management has evolved from a specialized computer application to a central component of virtually all enterprises, and, as a result, knowledge about database systems has become an essential part of an education in computer science. SQL is a powerful language for both querying and updating data in relational databases. Study of SQL empowers students to implement and work with relational data model.

2. Course Outcomes: At the end of the course, the student will be able to:

CO-01	Identify the elements of ER model for a given requirement, draw ER diagram and validate with the given requirement.
CO-02	Translate the given ER diagram to a relational model and verify against integrity constraints. Also refine and normalize the relational database design against first three normal forms.
CO-03	Use appropriate SQL statements to create a database and other DB objects using a DBMS software.
CO-04	Perform insert, delete and/or update operations on the database and query the database to retrieve the required information using appropriate SQL statements and clauses.

3. Course Content

Week	CO	PO	Lecture (Knowledge Criteria)	Tutorial (Activity Criteria)	Practice (Performance Criteria)
1	1	1,4	Introduction Overview of DB: why a database? Purpose of database; Classification; Application; DBMS: features, providers; Functional components of DBMS; Types of DBMS architecture; View of data in DBMS; Database users; Role and responsibilities of DBA; Case study: Example of any database application, recruitment database	Refer Table 1	1. Install and setup DBMS software such as MySQL, PostgreSQL 2. Learn the interface and explore the features of installed DBMS
2	1	1,3,4	Database design Data model; types; importance of data modeling; Overview of database design; phases		1. Identify and ER-model elements and draw ER

			of database design; database development life cycle; Conceptual design: ER-Model: entity: types; attribute: types; relationships: types, constraints, Symbols and Notations; Case study: conceptual design for a set of specifications <ul style="list-style-type: none"> i. Restaurant ii. Retail shop iii. Recruitment iv. College v. Library 		diagram for the given specifications using tools.
3	2	1,3	Relational model: Overview; characteristics; Constraints: types; Operations; Advantages and Disadvantages; applications; Design anomalies; Features of good DB design;		<ol style="list-style-type: none"> 1. Map ER Model to relational model 2. Identify various constraints
4	2	1,3	Functional dependency: overview, rules, types; Normalization: normalization process; importance of normalization; 1NF, 2NF, 3NF Sufficient examples to understand the concept		<ol style="list-style-type: none"> 1. Normalize the above design
5	3	1,4	database languages: types, commands/tasks in each type; Integrity constraints; MySQL/PostgreSQL: overview; features; datatypes; Standardization guidelines;		<ol style="list-style-type: none"> 1. Validate the above design against integrity constraints
6	3	1,3,4	Defining Data: DDL CREATE, ALTER, DROP different DB objects; Temporary tables: types, create and use; external tables; Managing constraints		<ol style="list-style-type: none"> 1. Use MySQL/PostgreSQL DDL statements to create database and other DB objects for above design
7	3,4	1,3,4	Insert, delete and update data Modifying data: UPDATE and DELETE Update anomalies; impact of constraints Querying of available data: SELECT; Aliases; sorting data: ORDER BY		<ol style="list-style-type: none"> 1. Perform single table and multi table insertion 2. Perform delete and update operations 3. Querying single table 4. Sort the result set of a query
8	4	1,4	filtering data: WHERE, AND, OR, row limiting clause, IN, BETWEEN, LIKE; Joining table: INNER JOIN, LEFT JOIN,		<ol style="list-style-type: none"> 1. Querying single table 2. Filtering data 3. query multiple tables with joins
9	4	1,4	Grouping data: Aggregate functions, GROUP BY, HAVING; Set operators: UNION, INTERSECT, MINUS;		<ol style="list-style-type: none"> 1. Queries that use set operators 2. Report aggregated data using group functions
10	4	1,4	Subqueries: Comparator operators; subqueries: Single Row Subqueries; Multiple Row Subqueries; correlated subqueries; EXISTS, NOT EXISTS, ANY, ALL, SOME;		<ol style="list-style-type: none"> 1. Write sub queries to retrieve information from the created database
11	4	1,4	Views: create, drop and update; realization of views based on single and multiple tables;		<ol style="list-style-type: none"> 1. Create view and query

			DCL: Controlling user access: Differentiating system privileges from object privileges; Granting privileges on tables		2. Create users and assign privileges for DB operations
12	4	1,4	PL/SQL: variables, datatypes; control statements (decision making); Stored procedures and Functions Concept; syntax and structure of store procedure; syntax and structure of functions; calling a function; Examples;		1. Create and execute store procedures 2. Create and execute functions
13	4	1,4	Managing and controlling transactions: Introduction of transaction, ACID properties; states of transaction; Transaction control; Overview of transaction management, using transaction control commands: COMMIT, ROLLBACK, SAVE POINT, SET TRANSACTION; sufficient examples;		1. Create and execute transactions 2. Call previously created store procedure or function in transaction
Total in hours			39	13	52

***PO = Program outcome as listed and defined in year 1 curriculum**

Table 1: Suggestive activities for tutorials (the list is only shared as an example and not inclusive of all possible activities for that course. Student and faculty are encouraged to choose activities that are relevant to the topic and the availability of such resources at their institution)

1	1. Identify the drawbacks of file system and how DB enables us to overcome them. Identify distinguishable features of each of DBMS available in the market.
2	1. Transform given n-ary relationship to binary relationship 2. Document the steps to create ER diagram. 3. Identify the components of ER model in the given requirements.
3	1. Document the steps to create logical design 2. Discuss and document Codd's 12 rules
4	1. Explore and document other normal forms
5	1. study and present the working of SQL optimizer 2. Learn and report optimization techniques
6	1. Learn and demonstrate use of DISTINCT, ALL, IS NULL; 2. Learn and present RIGHT JOIN;
7	1. Identify the advantages of Cascading Referential Integrity Constraints
8	1. Identify need of subqueries
9	1. identify the advantages and disadvantages of store procedure and functions,
10	1. presentation on the latest developments in research and industry related to this course
11	1. Learn and present need of scalar subqueries
12	1. Learn iterative statements in PL/SQL

13	1. Does Relational model support storage of unstructured data, if no, what are the alternatives to store unstructured data.
-----------	--

4. Reference:

Sl. No.	Description
1	Database System Concepts by Abraham Silberschatz, Henry F. Korth, S . Sudarshan
2	https://binaryterms.com/
3	https://beginnersbook.com/
4	https://www.oracletutorial.com/

5. Equipment/software list

Sl. No.	Particulars	Specification	Quantity
1	Computers		20
2	MySQL workbench/ or equivalent software; Lucid chart, draw.io		

(4th – SEMESTER)

Data Structures with Python

Subject code – CSE401

1. Rationale

Data structures are the techniques organizing data and of designing the algorithms for real-life projects. Knowledge of data structures is essential for software design and development. Learning data structures with Python offer flexibility and ease of programming with many built in data structures and libraries.

2. Course Outcomes: At the end of the Course, the student will be able to:

CO-01	Explain data structures types, list their applications.
CO-02	Apply the right Algorithm design strategies to solve a given problem.
CO-03	Choose the right data structure to develop solution to a given computing problem.
CO-04	Analyse space and time complexities of the algorithm used and plot a graph.

3. Course Content

Week	CO	PO	Lecture (Knowledge Criteria)	Tutorial (Activity Criteria)	Practice (Performance Criteria)
1	01	1, 2, 3	Introduction to Data Structures, operations, classification, Characteristics. Primitive types – primitive data structures, python examples. Non primitive types - Non primitive data structures, python examples. Linear and nonlinear data structures – with python examples. Introduction, Abstractions, Abstract Data Types, An Example of Abstract Data Type (Student, Date, Employee), Defining the ADT, Using the ADT, Implementing the ADT.	Refer Table 1	1. Python program to Use and demonstrate basic data structures. 2. Implement an ADT with all its operations.
2	01, 02, 04	1, 2, 3, 4, 7	Algorithm Analysis – Space Complexity, Time Complexity. Run time analysis. Asymptomatic notations, Big-O Notation, Omega Notation, Theta Notation.		1. Implement an ADT and Compute space and time complexities. 2. Implement above solution using array and Compute space and time complexities and compare two solutions.

3	01, 02, 04	1, 2, 3, 4, 7	<p>Algorithm design strategies:</p> <p>Brute force – Bubble sort, Selection Sort, Linear Search.</p> <p>Decrease and conquer - Insertion Sort.</p>		<ol style="list-style-type: none"> 1. Implement Linear Search compute space and time complexities, plot graph using asymptomatic notations. 2. Implement Bubble, Selection, insertion sorting algorithms compute space and time complexities, plot graph using asymptomatic notations.
4	01, 02, 04	1, 2, 3, 4, 7	<p>Divide and conquer - Merge Sort, Quick Sort, Binary search.</p> <p>Dynamic programming - Fibonacci sequence</p> <p>Backtracking – Concepts only (Implementation examples with recursion in week 9).</p> <p>Greedy – Concepts only.</p>		<ol style="list-style-type: none"> 1. Implement Binary Search using recursion Compute space and time complexities, plot graph using asymptomatic notations and compare two. 2. Implement Merge and quick sorting algorithms compute space and time complexities, plot graph using asymptomatic notations and compare all solutions. 3. Implement Fibonacci sequence with dynamic programing.
5	01, 02, 03, 04	1, 2, 3, 4,	<p>Linear(arrays) vs nonlinear (pointer) structures – Run time and space requirements, when to use what?</p> <p>Introduction to linked list, Examples: Image viewer, music player list etc. (to be used to explain concept of list), applications.</p>		1. Implement Singly linked list (Traversing the Nodes, searching for a Node, Prepending Nodes, Removing Nodes)
6	01, 02, 03, 04	1, 2, 3, 4,	<p>The Singly Linked List- Creating Nodes, Traversing the Nodes, searching for a Node, Prepending Nodes, Removing Nodes. Linked List Iterators.</p>		1. Implement linked list Iterators.
7	01, 02, 03, 04	1, 2, 3, 4,	<p>The Doubly Linked List, Examples: Image viewer, music player list etc. (to be used to explain concept of list).</p> <p>DLL node, List Operations – Create, appending nodes, delete, search.</p>		<ol style="list-style-type: none"> 1. Implement DLL. 2. Implement CDLL

			The Circular Linked List-Organization, List Operations – Appending nodes, delete, iterating circular list.		
8	01, 02, 03, 04	1, 2, 3, 4	Last In First Out (Stack) Data structures – Example: Reversing a word, evaluating an expression, message box etc. (to be used to explain concept of LIFO). The Stack implementation – push, pop, display. Stack Applications- Balanced Delimiters, Evaluating Postfix Expressions.		1. Implement Stack Data Structure. 2. Implement bracket matching using stack.
9	01, 02, 03, 04	1, 2, 3, 4,	Recursion. Properties of Recursion. Recursive functions: Factorials, RecursiveCall stack, The Fibonacci Sequence. How Recursion Works- The Run Time Stack. Recursive Applications- Recursive Binary Search, Towers of Hanoi.		1. Program to demonstrate recursive operations (factorial/ Fibonacci) 2. Implement solution for Towers of Hanoi.
10	01, 02, 03, 04	1, 2, 3, 4,	The First In First Out (Queue) Data structure – Example: Media player list, keyboard buffer queue, printer queue etc. (to be used to explain concept of FIFO). Implementing the Queue and its operations using Python List. Priority Queues, Implementation.		1. Implement Queue. 2. Implement priority queue
11	01, 02, 03, 04	1, 2, 3, 4,	The Tree data structure – Example: File explorer/Folder structure, Domain name server. Tree Terminologies, Tree node representation. Binary trees, Binary search trees, Properties, Implementation of tree operations – insertion, deletion, search, Tree traversals (in order, pre order and post order).		1. Implement Binary search tree and its operations using list.
12	01, 02, 04	1, 2, 3, 4,	Depth-first traversal Breadth-first traversal Tree applications: Expression evaluation.		1. Implementations of BFS. 2. Implementation of DFS.
13	01, 03, 04	1, 2, 3, 4,	Introduction to Hashing. Hashing - Perfect hashing functions. Hash table Hash Functions, Operations, Hash collision, Application.		1. Implement Hash functions.
Total in hours			39	13	52

***PO = Program outcome as listed and defined in year 1 curriculum**

Table 1: Suggestive activities for tutorials (the list is only shared as an example and not inclusive of all possible activities for that course. Student and faculty are encouraged to choose activities that are relevant to the topic and the availability of such resources at their institution)

Sl. No	Activity
1	Design a Data structure for handling Student Records- Designing a Solution, Implementation (Using Basic DS).

2	Design a Data structure for handling Student Records- Designing a Solution, Implementation (Using ADT).
3	Optimize your solution (Bubble sort, selection sort and Insertion sort)
4	Implement Radix sort.
5	Prepare report on nonlinear data structures.
6	Design and implement sparse matrix representation using linked list.
7	Design and implement simple application that require DLL data structure.
8	Implement and demonstrate evaluating postfix expression.
9	Presentation on run time stack.
10	Design and implement priority queue data structure.
11	Prepare a Report on balanced trees.
12	Implement expression evaluation tree.
13	Prepare a report on hashing and analyze time complexity.

4. Reference:

Sl. No.	Description
1	Data Structures and Algorithms using Python by Rance D. Necaise
2	Python Data Structures and Algorithms by Benjamin Baka
3	www.geeksforgeeks.com

5. Equipment/software list

Sl. No.	Particulars	Specification	Quantity
1	Computers		20
2	Python 3.6		20
3	Editor such as iPython, Jupyter, spider, PyCharm		20

Operating System and Administration

Subject code – CSE402

1. Rationale

The Operating System knowledge and skill is an integral part in the study of computer science. It provides the platform for all other application to run on the machine, thus knowledge of operating system and administration becomes indispensable for understanding computing environment. It is essential to have knowledge of operating system's services and utilities to develop, deploy and maintain the software and hardware. The students will also be skilled in operating system virtualization, to create and manage virtual computing environment.

2. Course Outcomes: At the end of the course, the student will be able to:

CO-01	Explain functions and services of an operating system.
CO-02	Create a virtual environment and configure it to meet a specific application requirement.
CO-03	Identify and use Linux commands to create and manage simple file processing operations, organize directory structures, and develop shell script to automate given simple task.
CO-04	Demonstrate the role and responsibilities of a Linux system administrator and analyse problems using suitable diagnostic tools and resolve issues.

3. Course Content

Week	CO	PO	Lecture (Knowledge Criteria)	Tutorial (Activity Criteria)	Practice (Performance Criteria)
1	1	1,7	Overview of Operating System, Need for OS, Structure, OS Types, Examples of OS (desktop and mobile) Dual mode operation, Kernel and microkernel, Functions of OS User interfaces; Corporate Vs Personal needs; Types of OS installation	Refer Table 1	1. Types of OS installation 2. Boot methods 3 . File System and formatting 4. Post installation tasks
2	1, 2	1, 7	Virtualization technology, working, types		1. Install and configure

			<p>Potentials and challenges of Virtualization, Virtual Machines, Containers.</p> <p>Linux Boot process.</p> <p>Linux command line - Interpreter, shell, CLI over GUI, Types of users- super and normal, Linux user manual.</p>		<p>virtual machine- Virtual box/VMware, VMware player station.</p> <p>2. Download and install a terminal emulator and connect Linux VM via TE(optional).</p> <p>Significance of man command.</p>
3	1,3	1, 7	<p>File system - Pathnames, File system structure and its description, navigating the file system.</p> <p>File types, attributes, Access Control List (ACL), Adding text to file.</p> <p>Pipes, File Comparison, Filters / Text Processing Commands.</p>	Refer Table 1	<p>File and Directory commands:</p> <ol style="list-style-type: none"> 1. Create and delete directories and files, File movement, copy commands, Pipes (named & unnamed) 2. Commands for viewing File, File comparison, File manipulation, Altering file permission, File compression and decompression. 3. Text processing commands.
4	1,	1,2,3,7	<p>Process Management – Process, daemon, process states, PCB; Process scheduling Queue</p> <p>Operations on Processes - Process creation, Process termination, Interprocess communication.</p> <p>Scheduling - Long term, short term, and medium term; Context switch; Different types of CPU schedulers (Basic concept), Process priority; debugging (system hang)</p>	Refer Table 1	<ol style="list-style-type: none"> 1. Linux commands related to process creation and management- system calls fork() and exec(); bg, fg, nohup, pkill, nice, top, ps; 2. cron and at commands to schedule tasks.
5	1,3	1,2,3 4, 7	<p>Process synchronization- critical section problem, Semaphores; Deadlock- System model, methods for handling deadlocks, deadlock prevention, avoidance, detection, recovery from deadlocks.</p> <p>Threads - Multithreading models, Threads, and processes.</p> <p>Types of threads - Kernel level and User level</p>		<ol style="list-style-type: none"> 1. Demonstration through videos. 2. Commands to exhibit thread concepts.
6	1	1,2,3 4, 7	<p>Memory management - Process address space, static vs dynamic linking and loading.</p>		<ol style="list-style-type: none"> 1. Demonstration through videos. 2. Commands to view memory consumption

			Swapping, Memory allocation, Fragmentation, Paging, Segmentation; Virtual memory, Demand paging, Page replacement algorithm (concept only)		
7	1	2, 3 4, 7	Shell Programming: Basics of shell programming, types of shell in Linux, Basic Shell scripts- Shebang or Hashbang, Input & Output, decision making and iterative scripts.		1. Write shell scripts to illustrate decision making and different types of iterations; Ex- to perform string operations; to perform file operations;
8	1	2, 3 4, 7	Automation of system tasks: Writing scripts to automate common tasks.	Refer Table 1	1. Illustrate automation of basic tasks like monitoring memory consumption, check remote servers' connectivity, etc., at different frequencies.
9	1	2, 3, 4	Network Management Network components- IP address, subnet mask, gateway. Network Interface management; Communication. Data transfer facilitation. Diagnosis and troubleshooting; Resource analysis.	Refer Table 1	1. Enable internet on Linux VM. 2. Test and manage network using following commands ifconfig, iwconfig, ethtool, arptwatch, bmon, telnet, ssh, sendmail, mailstats, w cURL, wget, ftp, rcp, scp, rsync, sftp. netstat, ping, traceroute, iftop, nload, ss. tcpdump, dstat.
10	2, 4	2, 4, 7	User authentication User and Group account management. Working on interface. Linux Directory Service - Account Authentication, what is LDAP and Active Directory? LDAP structure, working.	Refer Table 1	1. Work on user accounts useradd, passwd, userdel, usermod, groupadd, groupmod, gpasswd, groupdel; system-config. 2. OpenLDAP Installation 3. LDAP server and client configuration.
11	4	2, 3 4, 7	System monitoring, Log monitoring System maintenance, System information. System architecture, Linux Boot process and System run levels, System updates and repositories.		1. System monitoring commands top, df, dmesg, iostat 1, free, cat /proc/cpuinfo, cat/proc/meminfo; 2. Work on log directory - /var/log; 3. System maintenance commands- shutdown, reboot,

					halt, init. 4. System update & repositories- yum & rpm.
12	2,4	2, 3, 4, 7	Server setup: DNS- Introduction, Configuration, creating DNS zone, using DNS tools; FTP- Installation process, configuration and securing; setting up an Apache Web Server(<i>http</i>)	Refer Table 1	Install and configure: 1. DNS server with a domain name of your choice. 2. FTP server on LINUX and transfer files to demonstrate it's working. 3. Apache web server and create virtual hosts.
13	2,4	2, 3 4, 7	Storage management: Disk partition, formatting, mounting; Logical Volume Management (LVM)- Use of LVM, creating Volume groups, logical volume and disk mirroring, Extend Disk using LVM, Adding Swap Space Introduction to RAID – Hardware & Software, RAID levels.	Refer Table 1	1. Basic commands for storage partitions. 2. Install and configure LVM. 3. Add Disk and Create Standard & LVM Partition. 4. Add virtual disk and create a new LVM partition(pvcreate, vgcreate, lvcreate) 5. Extend disk using LVM
Total in hours			39	13	52

***PO = Program outcome as listed and defined in year 1 curriculum**

Table 1: Suggestive activities for tutorials (the list is only shared as an example and not inclusive of all possible activities for that course. Student and faculty are encouraged to choose activities that are relevant to the topic and the availability of such resources at their institution)

1	1. Compare features of different OS(windows, Linux, RTOS- Vxworks/android) 2. Study the evolution of OS to recognize the importance of current OS trends. 3. Explain the different flavors of LINUX.
2	1. Explain OS level virtualization and state its benefits. 2. Compare VMs and Containers 3. Identify the difference between hypervisors and Linux containers. 4. Comprehend the benefits of virtualization.
3	1. Compare ex2/ex3 filesystem attributes. 2. Discuss the file- mount and unmount system calls.
4	1. Compare Linux fork () and Windows createprocess () functions.
5	1. Study probable conditions for deadlock occurrence and how to overcome it. 2. Identify relationship between threads and processes. 3. Comprehend the differences between types of threads
6	1. Compare the features of swapping and paging.
7	1. Compare different Linux shells.
8	1. Write a cron job that runs all essential apps. on an hourly/ daily/weekly/monthly basis. (for ex. Executing Antivirus)
9	1. Compare static and DHCP IP addresses and check whether these can be switched over. 2. Study different options offered by Linux for package management.
10	1. Identify few alternatives to openLDAP and make a comparison.
11	1. Explore other network commands required for a sysadmin and interpret their functions and usage.

12	1. Study the difference between application server and web server. 2. Identify the role of virtual host. 3. Explain different types of Apache virtual hosts and how they are set up.
13	1. Compare the features between RAID and SSD.

4. Reference:

Sl. No	Description
1	Operating System internal and Design Principles, William Stallings
2	Operating System, Garry Nut
3	https://www.redhat.com/en/topics/virtualization
4	Virtual Machine - an overview ScienceDirect Topics
5	DNS: https://www.youtube.com/watch?v=TiWs9n4fhys&list=RDCMUCQSpnDG3YsFNf5-qHocF-WQ
6	Linux system admin requirements: https://www.temok.com/blog/linux-system-administration/
7	Linux commands for modern sysadmins- N/W related - https://www.ubuntupit.com/useful-linux-network-commands-for-modern-sysadmins/
8	DNS Technology: https://www.digitalocean.com/community/tutorials/an-introduction-to-dns-terminology-components-and-concepts
9	Commands for Disk Management: https://www.programmersought.com/article/55913754022/

5. Equipment/software list

Sl. No.	Particulars	Specification	Quantity
1	Computers		
2	VirtualBox, Ubuntu or any other Linux OS image.		

Object Oriented Programming and Design with Java

Subject code – CSE403

1. Rationale

Object oriented programming paradigm with object-oriented design principles are vital in design and development of today's complex computing solutions. OOD principles provide valuable standards and guidelines to create clean and modular design and avoid code smells. Java being the popular object-oriented programming language that empowers the innovation in this digital world, students will have sound knowledge of object-oriented programming concepts and design principles with java.

2. Course Outcomes: At the end of the course, the student will be able to:

CO-01	Design a solution for a given problem using object-oriented programming concepts and apply all appropriate object-oriented design principles
CO-02	Write and test the code for a designed solution using java OOP concepts.
CO-03	Identify exceptions in the designed or given solution and explain how to resolve them.
CO-04	Demonstrate with an example a java application's connection with a database.

3. Course Content

Week	CO	PO *	Lecture (Knowledge Criteria)	Tutorial (Activity Criteria)	Practice (Performance Criteria)
1	1,2	1, 4	Introduction to Java Brief history; features; java architecture; components: JVM, JRE, JDK; Applications; Java environment setup; Structure of java program; Compilation and execution of java program; Clean coding in java.	Refer Table 1	1. Install and Setup java environment 2. Install java editor (Eclipse for Enterprise Java) and configure workspace 3. Execution of first java program 4. Java code execution process
2	1,2	1, 2, 3, 4, 7	Introduction to OOP: Building blocks: class, object, attributes, methods; Class and objects in java;		1. Code, execute and debug programs that uses different types of variables and datatypes;

			Variable: Types (local, instance, static); declaration, initialization; comments; 'Data types;		2. Identify and resolve issues in the given code snippet
3	1,2	1, 2, 3, 4	Constructors: rules for defining constructor; types; Destructor; Access modifiers; this keyword; Autoboxing and unboxing; Operators; Expressions; Evaluation of expressions;		1. Code, execute and debug programs <ol style="list-style-type: none"> that uses different types of constructors for expression evaluation to perform autoboxing and unboxing 2. Identify and resolve issues in the given code snippet
4	1,2	1, 2, 3, 4, 7	Memory allocation in java; garbage collection: concept, working, types, advantages finalize () method;		1. Install memory monitoring tool and observe how JVM allocates memory 2. Memory allocation explanation through the programs
5	1,2	1, 2, 3, 4	Conditional and Iterative statements Decision making: if, if..else, switch Iterative: need of iterative statements; types of loops in java; how to use them; Break and continue statements;		1. Code, execute and debug programs that uses different control statements. 2. Identify and resolve issues in the given code snippet
6	1,2,3	1, 2, 3, 4, 7	OOP concepts: Encapsulation Concept; What is encapsulation? How to achieve encapsulation in java; Packages; Single Responsibility Principle: Intent; Rules; Benefits; example		1. Code, execute and debug programs 2. that uses encapsulation concept. 3. Define class & implement like simple calculator or text processing and check compliance with SRP.
7	1,2	1, 2, 3, 4	Arrays: Why arrays? Features, types, Declaration, array creation with new operator, working with arrays; Strings: creation, string methods;		1. Code, execute and debug programs that uses array concept 2. Code, execute and debug programs to perform string manipulation.
8	1,2	1, 2, 3, 4, 7	OOP concepts: Inheritance Inheritance concept; types; Inheritance in java; Examples; Open Closed principle: Intent; Rules; Benefits; example	Refer Table 1	1. Code, execute and debug programs that uses inheritance concept 2. Design a class & implement like file parser and check compliance with OCP.

9	1,2	1, 2, 3, 4,7	OOP concepts: Polymorphism Polymorphism concept; types: method overloading and overriding; application; polymorphism in java; sufficient examples;		1. Code, execute and debug programs that uses a. static binding b. dynamic binding
10	1,2	1, 2, 3, 4, 7	OOP concepts: Abstraction Overview; implementation of abstraction in java: abstract class and interface; Relationship between class and interface; inheritance in interface; Examples to substantiate the understanding of concepts; Eg. File parser; message logger		1. Code, execute and debug programs that uses 2. abstract class to achieve abstraction 3. interface to achieve abstraction 4. Verify whether the given code snippet is correct according to abstraction or not
11	1,2,3	1, 2, 3, 4, 7	Files and Exception handling Files and I/O streams: File reader and writer; Exception concept; exceptions in java; classification: checked and unchecked; exception handling in java;		1. Code, execute and debug programs in java to a. handles checked and unchecked exceptions b. read the content of the file and write the content to another file 2. Incorporate exception handling in programs/applications developed in previous sessions.
12	1,2,3	1, 2, 3, 4, 7	Design principle: Interface Segregation principle: Intent; Rules; Benefits; examples; Enums; Overview of java annotations;	Refer Table 1	1. Design an interface & implement it like one that builds different types of toys and check compliance with ISP.
13	1,2,3,4	1, 2, 3, 4, 7	Database Connectivity Introduction to JDBC; JDBC components; How JDBC works? JDBC connections; Connect java application to database using JDBC;		1. Code, execute and debug programs to connect to database through JDBC and perform basic DB operations
Total in hours			39	13	52

*PO = Program outcome as listed and defined in year 1 curriculum

Table 1: Suggestive activities for tutorials (the list is only shared as an example and not inclusive of all possible activities for that course. Student and faculty are encouraged to choose activities that are relevant to the topic and the availability of such resources at their institution)

Sl. No	Activity
--------	----------

1	1. Identify various java IDEs and identify differences between them. 2. Compare and contrast Java with Python
2	1. Study and present a. type casting in java b. what are command line arguments in java? c. java keywords and their usage
3	1. Compare and contrast a. method and constructor; b. constructor and destructor
4	1. Study and present how does bytecode work in java.
5	1. Present nesting of conditional and iterative statements considering a use case.
6	Identify advantages and disadvantages of a. Encapsulation. b. Inheritance c. Abstraction d. Polymorphism
7	Study and report a. java Arrays class their methods b. java String class their methods
8	Identify and document how these principles help to avoid code smells. a. SRP b. OCP c. ISP
9	Compare and contrast a. static and dynamic binding and identify usage of each b. abstract class and interface, identify usage of each
10	1. Differentiate error and exception 2. Identify and document system exceptions
11	Study DRY principle, identify the benefits.
12	Identify how OOD principles violations impact the quality of code.
13	Identify java ORM frameworks and their features.
14	Study and find the inclusions in latest java versions.

4. Reference:

Sl. No.	Description
1	https://docs.oracle.com/javase/tutorial/java/concepts/
2	www.edureka.co
3	Clean Code by Robert C Martin
4	https://www.javabrahman.com/programming-principles/
5	https://medium.com/

5. Equipment/software list

Sl. No.	Particulars	Specification	Quantity
1	Computers		
2	Java 8.0 and above, eclipse		

Software Engineering Principles and Practices

Subject code – CSE404

1. Rationale

Digital reality has become an integral part of human life with software tools being used to deal with virtually every part of life. A process is key to develop a quality software successfully. Principles and practices of software engineering blends engineering, computing, project management and software development. It's essential to understand the life cycle of software development and the process followed to develop a quality software. Design thinking methodology encourages identifying alternative strategies and solutions to solve a problem in best possible way.

2. Course Outcomes: At the end of the Course, the student will be able to:

CO-01	Explain the typical software development life cycle (SDLC), list and differentiate the various SDLC models along with identifying where each model could be beneficial when applied.
CO-02	Demonstrate the application of design thinking as a process, explain how it helps in requirement engineering and mitigate risks.
CO-03	Study a given application requirement, create user stories, draw the appropriate UML diagram and validate to ensure user story/UML diagram meet with the given requirement.
CO-04	Document standard test procedures and test cases for a given requirement to ensure the software gives the desired results for which it is built.

3. Course Content

Week	CO	PO	Lecture (Knowledge Criteria)	Tutorial (Activity Criteria)	Practice (Performance Criteria)
1	1	1	Overview Software engineering; Need of software engineering; Software paradigms; Software product types: generic, customized; characteristics of good software; Challenges in software projects; Factors that influence software development; understanding success Software process; need of process, components of process, process	Refer Table 1	<ol style="list-style-type: none">1. Discuss success and failure stories2. Presentation of collected case studies3. Enact the importance of ethical practices

			activities; Differentiate product, project and process; process assessment and improvement; Software engineering ethics.		
2	1	1, 5	SDLC and Process Models SDLC; Software process model; How to choose process model? Comparison between a defined process and an empirical process; Traditional process models: waterfall; Incremental; Agile process- manifesto; principles; practices; A paradigm shift from plan driven mentality.		1. Case study to understand the SDLC 2. Organize and play games to understand the agile process like, morning wake up game <ul style="list-style-type: none"> the marshmallow challenges White Elephant Sizing Easter Egg Challenge 3. Create JIRA (similar tool) account and learn interface <input type="checkbox"/>
3	1	1, 5	Agile frameworks; Ceremonies; Roles; Overview of XP – XP practices Scrum: Overview; framework; ceremonies and artifacts		1. Play and act agile ceremonies 2. Play different agile roles Eg. Product owner, business analyst
4	1,2	1	Risk Risk, characteristics, categories; why risk management is critical; risk management framework; Activities; Principles of risk management, Risk identification, Risk assessment – risk analysis; risk prioritization; Risk Mitigation; need and importance of risk mitigation; Risk Control – planning, resolution, monitoring; How to use tool to manage and mitigate risks in an organization.		1. case study to understand the importance of risk management and mitigation of risk 2. How to use tools to manage and mitigate risks [eg. Logicgate, AuditBoard etc]
5	2	1, 2	Design Thinking Introduction, 5 stages of design thinking Understand the process of design thinking using an example Case Study		1. Conduct warmup activities to Ignite Design Thinking 2. Organize and conduct design thinking exercises and games

6	1,3	1, 2	Requirement Engineering & Modeling Overview; what is requirement? Importance; Requirement types; Sources of requirements; Requirement engineering Process; Feasibility study; Typical Requirements Engineering Problems; Requirement modeling strategies; Overview of UML; types of diagrams; Note: Take a case study to understand requirement engineering and prepare use cases or user stories	1. Organize role play for requirement activities 2. Identify a problem and prepare requirement document or Epics and user stories 3. Configure JIRA for the managing the project to solve the identified problem 4. Draw UML diagram for given use case
7	1,3	1, 2, 4	User stories What are user stories? Why user story? Basic concepts; Characteristics; How to write/create user stories? Steps; 3C's in user stories; Life cycle of user story. User story map. Estimation: User story point: basics; components of story point estimation; Steps involved in estimation;	1. Create detailed user stories for the above identified problem 1. Organize and play planning poker to decide on user points.
8	1,2	1, 2, 3	Design Objectives; design Concepts; Levels of design; Architectural styles; Monolithic and Microservices; UI and UX: Overview of UI and UX, UI types, essential properties, elements of UI design; relationship between UI and UX; Importance of good UI/UX. Wireframes: overview, purpose, benefits;	1. Create sitemap and wireframe for above created user stories. (Tools such as sketch, Adobe XD, Figma, etc can be used)
9	1	1, 2, 3, 4	Development Overview of DevOps; working principle; Benefits; DevOps culture; DevOps practices: continuous integration, continuous delivery, version control, configuration management, Build process;	1. Create Git (similar tool) account and configure repository 2. Upload the artifacts created to Git Learn version control and configuration management with Git
10	1	1, 2, 3, 4	Code quality and code security: overview; importance; issues caused by poor code; tools to check code quality Containerization: Container, why container, containerization; working principle; benefits; Hello world example Note: Docker or similar tool can be used to explain the containers.	1. Install and configure Jenkins 2. Create a container image for Hello world project 3. Setup build for container image using Jenkins (Hello world application)

11	1,4	1,4	Testing Principles of testing; Need of testing; stages; Testing process and activities; classification; Testing strategies; Levels of software testing; Software testing types; (Integration testing, functional testing, end-to-end testing need to be explained in detail)		1. Prepare Test plan for the user stories using JIRA 2. Prepare RTM for the user stories created using JIRA. Create test cases for the user stories created.
12	1,4	1	Software Measurement and Metrics Measurement; need of Measurement; types; Metrics: characteristics; classification; Agile metrics; Application monitoring.		1. Use JIRA or similar tool to capture agile metrics 2. Use SonarQube to capture code quality metrics
13	1,4	1	Quality Control and Assurance Concept of software quality, Compliance, Quality Standards, quality control, quality assurance; Difference between QC & QA. Need for auditing. Auditing fundamentals: auditing, elements of auditing; audit types; auditing methods, benefits of auditing. Quality and Process improvement tools and techniques– pareto chart, PDCA cycle, Six sigma and Lean process		1. Organize Roleplay to understand the roles and responsibilities of QA and QC team. 2. Audit the artifacts produced in previous sessions
Total in hours			39	13	52

*PO = Program outcome as listed and defined in year 1 curriculum

Table 1: Suggestive activities for tutorials (the list is only shared as an example and not inclusive of all possible activities for that course. Student and faculty are encouraged to choose activities that are relevant to the topic and the availability of such resources at their institution)

Sl No	Activity
1	Study the traffic signal and the importance of rules and process.
2	Visit various consulting company web portals and collect case studies.
3	Document the roles and responsibilities of different agile ceremonies
4	Identify cost of risk; Identify commonly used risk management tools.
5	Identify a problem and explain how design thinking can be applied to solve it. Design a shopping cart to achieve ease of use, applying design thinking.
6	Prepare RPM requirement traceability matrix for shopping cart List the criteria to select the requirement management tools. Identify different requirement management tools and list their features. Identify frequently used UML diagrams and also identify tools used to draw them.
7	Explore agile estimation techniques and prepare a report.
8	Study boiler plate and present necessary characteristics of boiler plate for a large and small project
9	Identify different DevOps Tools and list their features Study and report OWASP coding guidelines Learn and report Twelve Factor App methodology Identify different version control and configuration management tools and report their marketshare

10	Compare and contrast containerization and virtualization and identify importance of these in software development Identify container providers
11	Study and prepare report on testing tools. Compare manual and automation testing
12	Study and prepare report on widely used software metrics.
13	Identify different quality tools and report their features and usage

4. Reference:

Sl. No.	Description
1	Agile Software Development, principles, patterns and practices by Robert Martin
2	Art of agile development by James Shore & Shane Warden
3	Extreme programming explained: embrace change
4	Software-Engineering-9th-Edition-by-Ian-Sommerville
5	RPL-7th_ed_software_engineering_a_practitioners_approach_by_roger_s._pressman_
6	Becoming Agile..in an imperfect world by Greg Smith, Ahmed Sidky
7	scaledagileframework.com
8	Continuous Delivery Principles Atlassian
9	www.agilealliance.org/
10	www.udemy.com
11	www. tutorialride.com
12	www.interaction-design.org/
13	www.digite.com

5. Equipment/software list

Sl. No.	Particulars	Specification	Quantity
1.	Computers		
2.	Git, Jira, SonarCube, Lucidchart or any other UML design tool		