Information Technology- Object-Oriented Programming (OOP) Mini-Project
Objective: To apply core object-oriented programming principles such as inheritance, polymorphism, and encapsulation in creating a basic application. This experiment helps students practice real-world coding skills in OOP languages like Java, Python, or C++.

Materials Needed:

- Computer with IDE (e.g., IntelliJ IDEA, PyCharm, Visual Studio)
- Programming language: Java, Python, or C++
- Design tools (e.g., UML for class design)

Procedure:

- Define a project idea (e.g., a simple library management system, a student record system).
- Break the project into classes and define their attributes and methods.
- Implement inheritance where a subclass inherits properties from a parent class (e.g., "Book" class can be inherited by "FictionBook" and "NonFictionBook").
- Apply polymorphism to create methods that can operate on different types of objects in a flexible way.
- Use encapsulation to hide object data and provide access through getter and setter methods.
- Test the program with sample inputs and outputs, ensuring that all OOP principles are correctly implemented.

Information Technology- Data Visualization Using Python
Objective: To create visual representations of data through charts and graphs, helping students to analyze and present data effectively. This experiment introduces data visualization using Python libraries like Matplotlib and Seaborn.

Materials Needed:

- Python environment (e.g., Anaconda, Jupyter Notebooks)
- Pandas, Matplotlib, Seaborn libraries
- Sample datasets (e.g., sales data, weather data)

Procedure:

- Import a dataset into Python using the Pandas library.
- Clean and preprocess the data, handling missing values and formatting issues.
- Use Matplotlib or Seaborn to create line plots, bar charts, and pie charts to represent the data visually.
- Explore advanced visualization techniques like heatmaps, scatter plots, and pair plots.
- Create a dashboard using Streamlit to present interactive visualizations.
- Present the insights from the visualizations, discussing trends and patterns observed in the data.

Information Technology- Cybersecurity – Password Cracking Demo
Objective: To understand the vulnerabilities of password security and the techniques used to crack passwords. This experiment emphasizes the importance of strong password policies.

Materials Needed:

- Computers with cybersecurity software (e.g., Hashcat, John the Ripper)
- Sample password hashes (SHA-256 or MD5)
- Dictionaries for brute-force attack simulation

Procedure:

- Create sample password hashes using a hash function like SHA-256 or MD5.
- Use Hashcat or John the Ripper to perform dictionary and brute-force attacks on the hashed passwords.
- Observe how long it takes to crack passwords based on their complexity and length.
- Compare the effectiveness of different attack methods, discussing how stronger password policies can mitigate these risks.
- Recommend best practices for creating strong passwords and securing sensitive information.

Information Technology- API Integration Lab
Objective: To fetch and process real-time data using external APIs, allowing students to integrate external resources into their applications.

Materials Needed:

- Python (with requests library) or JavaScript (with Fetch API)
- Access to public APIs (e.g., weather API, news API)
- IDE or text editor (e.g., VS Code)

Procedure:

- Select a public API (e.g., OpenWeatherMap, NewsAPI) and obtain an API key.
- Write code in Python or JavaScript to fetch data from the API using HTTP requests.
- Parse the JSON response and extract useful information (e.g., weather conditions, news headlines).
- Display the fetched data in a simple user interface (UI) or output format.
- Handle common API issues like rate limits, timeouts, and errors in the response.
- Discuss the practical applications of APIs in web development and real-time data integration.

Information Technology- Linux Command Line & Shell Scripting
Objective: To automate routine tasks and gain proficiency with the Linux command line. This experiment teaches students shell scripting basics for task automation.

Materials Needed:

- Linux OS (e.g., Ubuntu)
- Text editor (e.g., nano, Vim)
- Terminal for command execution

Procedure:

- Write shell scripts in Bash to automate tasks like file management, backups, or system monitoring.
- Use common Linux commands such as grep, awk, sed, cp, mv, tar, and chmod.

- Set up cron jobs to schedule periodic tasks.
- Test the script with real-world examples (e.g., automatic backup of files or logs).
- Evaluate the efficiency of the script and discuss how shell scripting can simplify system administration tasks.

Information Technology- WebSocket Chat Application
Objective: To build a real-time messaging app using WebSockets, introducing students to web communication protocols and interactive applications.

Materials Needed:

- Node.js and npm
- Socket.IO library for WebSocket communication
- HTML/CSS/JavaScript for front-end

Procedure:

- Set up a Node.js project and install the Socket.IO library.
- Create a WebSocket server to handle incoming messages from clients.
- Develop a simple front-end with HTML and CSS for the chat interface.
- Use JavaScript and Socket.IO to establish a real-time connection between clients and the server.
- Test the application with multiple users and implement features like user notifications and message history.
- Analyze how WebSockets differ from HTTP and discuss their applications in real-time systems.

Information Technology- Machine Learning – Image Classifier
Objective: To train a simple machine learning model that can classify images into predefined categories. This experiment introduces students to deep learning, convolutional neural networks (CNN), and the basics of training and evaluating models.

Materials Needed:

- Python (with libraries: TensorFlow, Keras, NumPy, matplotlib)

- Dataset (e.g., MNIST, CIFAR-10)
- Jupyter Notebook or any IDE that supports Python
- GPU (optional, for faster training)

Procedure:

1. Import the necessary libraries (TensorFlow/Keras, NumPy, matplotlib).
2. Load and preprocess the dataset (e.g., MNIST or CIFAR-10), which includes normalizing pixel values and splitting data into training and testing sets.
3. Design a convolutional neural network (CNN) model with layers like Conv2D, MaxPooling2D, and Dense.
4. Compile the model using an appropriate optimizer (e.g., Adam) and loss function (e.g., categorical cross-entropy).
5. Train the model on the training data and validate it on the test data.
6. After training, evaluate the model's performance using metrics such as accuracy and loss.
7. Plot graphs for accuracy and loss over epochs to assess model convergence.
8. Test the trained model by classifying new images from the dataset and comparing the results with true labels.
9. Discuss methods to improve model performance, such as adding more layers, data augmentation, or hyperparameter tuning.

Information Technology- DevOps CI/CD Pipeline
Objective: To automate the process of code integration, testing, and deployment using a Continuous Integration/Continuous Deployment (CI/CD) pipeline. This experiment provides hands-on experience in automating software delivery and ensuring code quality.

Materials Needed:

- GitHub or GitLab repository
- Jenkins, GitHub Actions, or GitLab CI for pipeline automation
- Docker (optional, for containerization)
- Python or Node.js application (for testing)

- Text editor (VSCode or similar)

Procedure:

1. Set up a GitHub repository and push a basic web application or script (e.g., Python Flask app, Node.js app).
2. Choose a CI/CD tool (e.g., Jenkins, GitHub Actions) and set up a pipeline configuration file (YAML or Jenkinsfile).
3. Configure the pipeline to automatically build the project whenever new code is pushed to the repository.
4. Integrate automated testing in the pipeline by running unit tests or integration tests after the build stage.
5. Set up a deployment stage that automatically deploys the application to a staging server after successful tests.
6. Monitor the pipeline's execution in the CI/CD tool and ensure all stages are completed successfully.
7. Test the application after deployment to ensure the system is working as expected.
8. Discuss potential improvements to the pipeline, such as adding security scanning or rolling deployment strategies.

Information Technology- Cloud Computing – AWS EC2 Deployment
Objective: To deploy a web application on a cloud platform using Amazon Web Services (AWS) EC2 instances. This experiment teaches students about cloud infrastructure, virtual machines, and hosting a scalable application.

Materials Needed:

- AWS account (free tier for students)
- Flask or Node.js application
- EC2 instance (t2.micro or similar)
- SSH key pair for EC2 access
- Terminal/Command Prompt for SSH

Procedure:

1. Sign in to AWS and launch a new EC2 instance with a supported operating system (Ubuntu or Amazon Linux).
2. Configure the security group to allow HTTP (port 80) and SSH (port 22) access.
3. Generate an SSH key pair to access the EC2 instance securely.
4. SSH into the EC2 instance using a terminal and the generated key pair.
5. Install necessary software (e.g., Apache, Nginx, Python, or Node.js) on the EC2 instance.
6. Upload your application files (e.g., a Flask or Node.js web app) to the EC2 instance.
7. Start the application on the EC2 instance and configure the web server (Apache/Nginx) to serve the application.
8. Access the web app through the public IP address of the EC2 instance to verify deployment.
9. Discuss cloud scalability options, such as using Elastic Load Balancers or AWS Auto Scaling.

Information Technology- Blockchain Smart Contract Lab
Objective: To write, deploy, and interact with a simple blockchain smart contract on the Ethereum test network using Solidity. This experiment introduces students to the fundamentals of blockchain development and decentralized applications (DApps).

Materials Needed:

- Ethereum testnet (Rinkeby, Ropsten, or Kovan)
- Solidity programming language
- Remix IDE (for smart contract development)
- MetaMask (for interacting with the contract)
- JavaScript (for frontend integration)

Procedure:

1. Install MetaMask browser extension and set up an Ethereum wallet connected to a test network.

2. Access Remix IDE (web-based Solidity editor) and write a simple smart contract (e.g., a contract that manages a token or performs simple operations like a voting system).
3. Compile the contract in Remix IDE and deploy it to the Ethereum testnet using your MetaMask wallet.
4. Verify the deployment by interacting with the contract using Remix's interface or through a JavaScript frontend connected to MetaMask.
5. Test the functions of the contract (e.g., transfer tokens, vote, or store data) and observe the results on the blockchain.
6. Monitor transactions on the blockchain explorer for the respective test network (e.g., Rinkeby).

Discuss the benefits and challenges of deploying smart contracts on the mainnet, including gas fees, security, and scalability.

Architecture- Climate-responsive Architecture Study

Objective: To analyze the thermal performance of different building envelope materials and orientations based on local climate conditions. Students will understand how passive design strategies such as orientation and material selection can help optimize thermal comfort and energy efficiency in buildings.

Materials Needed:

- Climate simulation software (e.g., ClimateStudio, DesignBuilder)
- Computer with CAD software for modeling
- Local climate data (temperature, humidity, wind speed)
- Reference building model with adjustable orientations and materials

Procedure:

- Model a basic house using the simulation software, ensuring to include different orientations (North, South, East, West).
- Input local climate data (temperature, solar radiation, wind patterns) into the simulation.
- Use various building envelope materials (brick, insulated concrete, timber, glass) to simulate their thermal performance.
- Experiment with different window-to-wall ratios and placement for maximum passive solar gain.
- Run the simulation for various seasons, tracking interior temperature changes.
- Analyze the results and compare the impact of different materials and orientations on indoor temperature regulation.
- Report on the most effective passive design strategies for minimizing energy consumption and maximizing thermal comfort.

Architecture- Daylighting Simulation

Objective: To evaluate the natural light penetration into a building interior using various window sizes and orientations. Students will assess how to optimize daylighting for energy efficiency and user comfort.

Materials Needed:

- Daylighting software (e.g., DIALux, Velux Daylight Visualizer)
- CAD model of a building or room
- Local geographic data (latitude, longitude)
- Different window materials and sizes for simulation

Procedure:

- Create a 3D model of a room or building using CAD software and simulate different window placements.
- Input local geographic data and adjust the model's orientation to reflect seasonal variations in daylight.
- Use daylighting software to simulate light levels inside the room at various times of day and year.
- Measure lux values at multiple points in the room and assess uniformity of lighting.
- Test the impact of varying window sizes and glazing types on daylight penetration.
- Analyze the results to determine the most efficient daylighting strategy, suggesting improvements for energy savings and comfort.

Architecture- Building Material Load Testing
Objective: To examine the structural behavior of various building materials under compressive loads. This experiment teaches students about material strength and how different materials can impact structural integrity.

Materials Needed:

- UTM (Universal Testing Machine)
- Samples of materials (clay blocks, AAC blocks, concrete blocks)
- Digital caliper or micrometer
- Compression testing equipment (strain gauges, load cells)

Procedure:

- Prepare test specimens from three different types of building blocks: clay, AAC, and concrete.
- Measure and record dimensions of each specimen (length, width, height) using digital calipers.
- Place each sample in the UTM and apply gradual compressive loads.
- Continuously monitor deformation using strain gauges attached to the sample.
- Record the load at which cracking or failure occurs for each material.
- Plot a load vs. deformation curve for each material.
- Compare the compressive strength and behavior under load of each material, discussing which is most suitable for various building applications.

Architecture- Sound Absorption Lab
Objective: To measure and analyze the acoustic properties of different interior materials. This experiment helps students understand soundproofing and how materials influence sound control in buildings.

Materials Needed:

- Sound meter (decibel meter)
- Various acoustic materials (foam, cork, gypsum, fiberglass)
- A test booth or small enclosed space
- Tone generator or speaker
- Reflective surface to amplify sound

Procedure:

- Prepare test booths with different interior materials like foam, cork, gypsum, and fiberglass.
- Use a tone generator or speaker to produce a constant sound frequency inside the booth.

- Measure the reverberation time (RT60) and sound absorption coefficient using a decibel meter.
- Vary the material placements and record the effects on sound levels and reverberation.
- Analyze the reverberation time and propose the best material combinations for spaces like recording studios or theaters.
- Discuss the relationship between material porosity, density, and acoustic absorption.

Architecture- Slope and Site Analysis
Objective: To perform a topographic analysis of a site and analyze slope gradients, assessing the suitability for construction. Students will learn how to conduct a site assessment to guide architectural design decisions.

Materials Needed:

- Topographic maps or digital elevation models (DEMs)
- Digital elevation tools (e.g., laser distance meters)
- Calculators or software for slope analysis
- Graph paper or CAD software for drafting

Procedure:

- Obtain a topographic map or DEM of the site to be analyzed.
- Use tools like laser distance meters to measure elevations at various points across the site.
- Calculate slope gradients at different points using the rise/run formula.
- Identify areas with steep gradients (greater than 15%) where construction may be impractical.
- Use CAD software or graph paper to mark the buildable zones and non-buildable zones.
- Propose cut-and-fill strategies to make the site suitable for construction while maintaining environmental sustainability.

Architecture- Wind Flow Model Study
Objective: To evaluate wind patterns around buildings and assess the impact of building design on natural ventilation. This experiment helps students understand how building shapes, orientations, and placement can influence air movement, comfort, and energy efficiency.

Materials Needed:

- Scale model of buildings (made from foam, cardboard, or 3D-printed materials)
- Smoke machine or a fan to simulate wind flow
- Wind tunnel or open space with airflow for testing
- Anemometers to measure air velocity
- Smoke paper or dye to visualize wind patterns

Procedure:

1. Construct a scale model of several buildings, including different architectural features like atriums, windows, and shading devices.
2. Set up a controlled wind flow using a fan or wind tunnel to simulate real-life wind conditions.
3. Position the buildings in various orientations (north, south, east, west) to test the effect of placement on wind patterns.
4. Use a smoke machine or smoke paper to visualize the airflow around the buildings.
5. Use anemometers to measure airflow speeds at different points around the buildings.
6. Observe areas of high or low pressure, turbulence, and wind eddies, which can affect comfort, ventilation, and energy usage.
7. Based on observations, recommend design adjustments to optimize natural ventilation and improve building performance.

Architecture- Concrete Mix Proportioning
Objective: To prepare concrete of different grades (M20, M25, M30) and test its workability,

strength, and durability. This experiment introduces students to the principles of mix design and the effects of various ingredients on concrete properties.

Materials Needed:

- Cement (Portland or specific type)
- Fine aggregates (sand)
- Coarse aggregates (gravel or crushed stone)
- Water
- Measuring containers and scales
- Slump test apparatus
- Concrete cube molds
- Universal testing machine (UTM) for compressive strength testing
- Curing tanks or containers

Procedure:

1. Select concrete grades (M20, M25, M30) and prepare corresponding mix ratios for cement, sand, aggregates, and water (e.g., M20 – 1:1.5:3, M25 – 1:1:2).
2. Measure the ingredients accurately based on the required mix design.
3. Mix the materials thoroughly in a concrete mixer until the mixture is uniform.
4. Perform the slump test to measure workability. Record the slump value and compare it with the desired consistency for each grade.
5. Pour the mixed concrete into cube molds and compact it using a tamping rod.
6. Cure the concrete cubes by keeping them in a water-filled tank for 7 days.
7. After curing, remove the cubes and test their compressive strength using a Universal Testing Machine (UTM).
8. Record the results, plot the strength vs. age curves, and compare the results with the designed mix properties.

Architecture- Model Urban Traffic Simulation
Objective: To study vehicular flow and pedestrian movement at intersections or plazas and

propose design modifications to reduce congestion. This experiment emphasizes the importance of urban design in transportation efficiency and pedestrian safety.

Materials Needed:

- Scale model of a city intersection or plaza
- Toy vehicles and pedestrian figures
- Traffic simulation software (e.g., PTV VISSIM or SimTraffic)
- Stopwatch and measuring tools
- Graph paper for sketching redesigns

Procedure:

1. Create a scaled physical model of an urban intersection or public plaza with roads, sidewalks, and key traffic points (e.g., entry/exit, pedestrian crossings).
2. Introduce toy vehicles and pedestrians into the model and observe the flow of traffic during peak and off-peak times.
3. Collect data on traffic delays, congestion, pedestrian crossings, and waiting times.
4. Use simulation software to replicate the scenario digitally, inputting factors like traffic light timings, road widths, and pedestrian paths.
5. Measure the vehicle and pedestrian flow, and note any bottlenecks or areas of inefficiency.
6. Suggest potential changes to road layout, pedestrian areas, or traffic control systems (e.g., redesigning intersections, adding more lanes, improving signals).
7. Present the findings and design proposals to improve traffic efficiency and safety.

Architecture- Rainwater Harvesting Design
Objective: To design a rainwater harvesting system for a residential or commercial building, assessing the potential for water conservation and sustainability. This experiment teaches students how to calculate water catchment potential and design filtration and storage systems.

Materials Needed:

- Roof area measurement tools (tape measure, calculator)
- Local rainfall data (monthly or annual averages)
- Storage tank design templates
- Pipe, filter, and storage materials
- Calculation sheets for water harvesting potential

Procedure:

1. Measure the roof area of a building or residential plot, noting the dimensions and shape to calculate the surface area.
2. Gather local rainfall data, including average monthly or annual precipitation rates.
3. Calculate the potential volume of rainwater that can be harvested using the formula:

Harvested Water (liters)=Roof Area (m^2)×Rainfall (mm)×Efficiency Factor

where the efficiency factor accounts for losses due to evaporation, overflow, etc.

4. Design a storage system (e.g., water tank) based on the calculated volume. Consider space constraints and aesthetic factors.
5. Select appropriate filtration materials (e.g., mesh filters, sand filters) to ensure water quality.
6. Draft plumbing layouts for the rainwater collection system, including gutters, pipes, filters, and storage.
7. Evaluate the cost, environmental benefits, and feasibility of the system.

Architecture- Digital Architectural Portfolio Development
Objective: To curate and present architectural work in a digital format, honing skills in graphic design, layout, and professional presentation. This experiment encourages students to create a polished portfolio that effectively communicates their design work.

Materials Needed:

- Computer with graphic design software (e.g., Adobe InDesign, Canva, PowerPoint)
- Digital copies of architectural drawings, renders, photographs, and design sketches
- Scanner or camera (if physical copies of work need digitization)

Procedure:

1. Collect and organize all relevant architectural work, including floor plans, sections, elevations, renders, sketches, and photographs of built projects.
2. Choose a digital design tool (e.g., InDesign, Canva, PowerPoint) and create a template that reflects a professional and visually appealing layout.
3. Arrange the materials thematically, categorizing the portfolio by design project, type of work (e.g., sketches, CAD drawings, 3D models), or a specific theme (e.g., sustainability, urbanism).
4. Ensure that each section has a clear title, description, and context (e.g., project brief, location, materials used).
5. Include high-quality images with minimal text, focusing on clean layouts that highlight the work.
6. Include a brief biography, contact information, and a statement of design philosophy at the end of the portfolio.

Present the portfolio in a group setting for critique and discussion. Provide constructive feedback to peers, focusing on design and presentation quality.

Industrial Engineering- Line Balancing Simulation
Objective: To distribute tasks across workstations to maximize throughput and minimize idle time. Students will learn to optimize production processes in manufacturing settings.

Materials Needed:

- Line balancing software (e.g., Simul8, FlexSim) or Excel
- Data on task times and precedence relationships
- Simulation tool or physical items for task representation

Procedure:

- Define a production process with multiple tasks and assign task times based on real-world data.
- Create a precedence diagram to define the order in which tasks must be performed.
- Use line balancing software or Excel to assign tasks to different workstations, aiming for balanced workloads and minimized idle time.
- Analyze the cycle time and throughput of the line.
- Adjust task assignments and re-test to achieve optimal production efficiency.
- Discuss the impact of load imbalances and propose potential solutions for improvement.

Industrial Engineering- Assembly Line Time and Motion Study
Objective: To evaluate the time efficiency and motion of workers on an assembly line. Students will analyze repetitive tasks to identify opportunities for performance improvement.

Materials Needed:

- Stopwatches or time-tracking devices
- Assembly materials (e.g., mock products, tools)
- Video camera or motion-capture system

Procedure:

- Set up a mock assembly line with various stations (e.g., screwing, packaging).
- Record the time taken for each task in the assembly process using stopwatches.

- Use video footage to perform a motion analysis of workers performing tasks, identifying any unnecessary motions or inefficiencies.
- Apply time-and-motion study techniques, such as Therbligs, to categorize movements and identify waste.
- Propose changes to reduce non-value-added motions and improve overall efficiency on the assembly line.

Industrial Engineering- Warehouse Layout Optimization
Objective: To design an efficient warehouse layout that minimizes travel time for goods retrieval. This experiment focuses on applying warehouse design principles for optimized space utilization.

Materials Needed:

- Warehouse simulation software (e.g., AnyLogic, FlexSim)
- Graph paper or CAD software for layout design
- ABC analysis data for inventory items

Procedure:

- Perform an ABC analysis on the warehouse inventory to categorize items into three classes (A: high value/fast-moving, B: moderate value/moderate-moving, C: low value/slow-moving).
- Use warehouse simulation software or graph paper to design a layout that optimizes item retrieval based on their ABC classification.
- Minimize travel distance by positioning high-demand items (Class A) closer to the picking area and less frequently picked items (Class C) further away.
- Simulate the process of storage and retrieval to evaluate throughput.
- Analyze the efficiency of the design and propose adjustments for further improvement.

Industrial Engineering- Ergonomic Evaluation of Tools
Objective: To compare the ergonomic performance of two different tools and evaluate their

impact on user comfort and efficiency. This experiment will help students understand how ergonomics can reduce fatigue and improve productivity.

Materials Needed:

- Two different types of tools (e.g., screwdrivers, pliers)
- Hand dynamometer or muscle fatigue measurement devices
- Survey forms for user comfort assessment
- Stopwatch for task timing

Procedure:

- Select two tools with different ergonomic features (e.g., grip, weight, handle design).
- Ask participants to perform a simple task using each tool (e.g., tighten screws).
- Measure muscle fatigue using a hand dynamometer at regular intervals.
- Record the time it takes to complete the task with each tool.
- Have users complete a survey rating comfort and usability.
- Compare results and provide recommendations for ergonomic improvements.

Industrial Engineering- Forecasting Lab Using Time Series
Objective: To apply statistical methods for demand forecasting based on historical data. This experiment introduces students to time series analysis and its practical application in inventory management.

Materials Needed:

- Historical sales or demand data
- Spreadsheet software (e.g., Excel) or statistical software (e.g., R, Python)
- Graphing tools for visualization

Procedure:

- Collect historical sales data for a product over a specific period.
- Use statistical techniques such as moving averages or exponential smoothing to create a forecast for future demand.
- Plot the original data and forecast on a time series graph to visualize trends.

- Calculate forecast accuracy using metrics like Mean Absolute Percentage Error (MAPE).
- Discuss the limitations of the forecasting model and suggest ways to improve its accuracy.

Industrial Engineering- Control Chart Construction
Objective: To monitor the stability of a process over time using control charts. This experiment allows students to understand process variation and make data-driven decisions for quality control.

Materials Needed:

- Data collection sheets (or software like Minitab)
- Control chart templates (Excel or specific software)
- Calculators for control limit determination (X-bar, R, p)

Procedure:

- Collect data from a manufacturing or service process, either variable (e.g., time, length) or attribute (e.g., defect counts).
- Calculate the mean and standard deviation for each sample.
- Create a control chart in Excel or specialized software (e.g., Minitab) for the data type (X-bar, R, or p chart).
- Plot the data points and calculate control limits based on the process's historical data.
- Interpret signals and trends, such as out-of-control conditions or patterns, to suggest corrective actions.
- Analyze the process stability and discuss improvements based on findings.

Industrial Engineering- Workstation Design Simulation
Objective: To apply anthropometric data to design a workstation that maximizes comfort and efficiency. This experiment helps students learn how to adapt workstation design to human factors.

Materials Needed:

- Anthropometric data (e.g., chair height, desk height, reach distances)
- CAD software (e.g., SolidWorks, AutoCAD)
- Ergonomics reference material (e.g., books or standards)

Procedure:

- Collect anthropometric data for various user profiles (e.g., height, arm reach, hand size).
- Using CAD software, design a workstation (desk, chair, keyboard, etc.) that accommodates the target user group.
- Simulate ergonomic performance by adjusting seat heights, desk angles, and tool placements based on the collected data.
- Validate the design using virtual prototypes or physical mockups, testing for comfort and accessibility.
- Evaluate the fit of the workstation using anthropometric percentile ranges (e.g., 5th, 50th, and 95th percentiles) and adjust designs as needed.

Industrial Engineering- Value Stream Mapping (VSM)
Objective: To identify value-added and non-value-added activities in a process and visualize them using value stream mapping (VSM). This helps students optimize workflow and reduce waste in production processes.

Materials Needed:

- Process data (time, cycle time, inventory)
- VSM templates (or software like Microsoft Visio)
- Pen, ruler, and graph paper (if done manually)

Procedure:

- Select a specific process, either in manufacturing or service, to map.
- Document the current state of the process, identifying each step, time taken, inventory levels, and resources used.
- Create a value stream map, using symbols to indicate different stages of the process (e.g., supplier, production, customer).

- Identify non-value-adding steps, bottlenecks, and delays.
- Propose future-state value stream mapping to eliminate waste, improve flow, and optimize process efficiency.

Industrial Engineering- Layout Design Using SLP and BLOCPLAN
Objective: To develop an efficient facility layout using Systematic Layout Planning (SLP) and BLOCPLAN software, optimizing for space, workflow, and accessibility.

Materials Needed:

- SLP templates or BLOCPLAN software
- Data on interdepartmental relationships and space requirements
- Computer with layout design software

Procedure:

- Define relationships between departments based on the flow of materials or information.
- Create an activity relationship chart (REL chart) to quantify interaction levels between departments (e.g., high, medium, low).
- Use SLP or BLOCPLAN software to generate potential layout options, considering factors such as space utilization, proximity of related departments, and ease of movement.
- Evaluate the efficiency of different layouts by calculating space efficiency and workflow patterns.
- Suggest the most effective layout and explain how it will improve operational performance.

Industrial Engineering- Lean Manufacturing Lab – 5S Implementation
Objective: To apply the 5S methodology in a real or simulated workspace to improve organization, cleanliness, and efficiency. This experiment teaches students the basics of lean manufacturing and waste reduction.

Materials Needed:

- Workspace (real or simulated)

- Labels, bins, and organizing materials for sorting
- Cleaning tools and supplies
- Camera or documentation for before-and-after comparisons

Procedure:

- Start by photographing the existing setup of the workspace.
- Apply the 5S steps: **Sort** (eliminate unnecessary items), **Set in Order** (organize items for ease of use), **Shine** (clean the workspace), **Standardize** (set procedures for maintenance), and **Sustain** (implement habits to maintain improvements).
- Measure improvements in efficiency, space utilization, and worker satisfaction.
- Document the changes with photographs and statistics, comparing before-and-after results.
- Present findings and discuss the impact of 5S on workplace productivity and safety.

2.2.13 Cisco Packet Tracer – Point-to-Point Single-Area OSPFv2 Configuration

Objectives

- Part 1: Configure Router IDs.
- Part 2: Configure Networks for OSPF Routing.
- Part 3: Configure Passive Interfaces.
- Part 4: Verify OSPF Configuration.

Background

In this activity, you will activate OSPF routing using network statements and wildcard masks, configure OSPF routing on interfaces, and use network statements with quad-zero masks. In addition, you will configure explicit router IDs and passive interfaces.

Instructions

Part 1: Configure Router IDs

a. Start the OSPF routing process on all three routers. Use process ID 10.

Router(config)# router ospf 10

b. Use the router-id command to set the OSPF IDs of the three routers:

R1(config)# router ospf 10

R1(config-router)# router-id 1.1.1.1

R2(config)# router ospf 10

R2(config-router)# router-id 2.2.2.2

R3(config)# router ospf 10

R3(config-router)# router-id 3.3.3.3

Part 2: Configure Networks for OSPF Routing

Step 1: Configure networks on R1 using network statements and wildcard masks

R1(config-router)# network 192.168.10.0 0.0.0.255 area 0

R1(config-router)# network 10.1.1.0 0.0.0.3 area 0

R1(config-router)# network 10.1.1.4 0.0.0.3 area 0

Step 2: Configure networks on R2 using interface IP addresses and quad-zero masks

R2(config-router)# network 192.168.20.1 0.0.0.0 area 0

R2(config-router)# network 10.1.1.2 0.0.0.0 area 0

R2(config-router)# network 10.1.1.9 0.0.0.0 area 0

Step 3: Configure OSPF routing directly on interfaces of R3

R3(config)# interface GigabitEthernet0/0/0

R3(config-if)# ip ospf 10 area 0

R3(config)# interface Serial0/1/0

R3(config-if)# ip ospf 10 area 0

R3(config)# interface Serial0/1/1

R3(config-if)# ip ospf 10 area 0

Part 3: Configure Passive Interfaces

Configure the OSPF process on each of the three routers to make their LAN interfaces passive.

R1(config)# router ospf 10

R1(config-router)# passive-interface GigabitEthernet0/0/0

R2(config)# router ospf 10

R2(config-router)# passive-interface GigabitEthernet0/0/0

R3(config)# router ospf 10

R3(config-router)# passive-interface GigabitEthernet0/0/0

Part 4: Verify OSPF Configuration

Use the appropriate show commands to verify the network and passive interface configuration of the OSPF process on each router.

2.3.11 Cisco Packet Tracer – Determine the DR and BDR

Objectives

- Part 1: Examine DR and BDR Changing Roles
- Part 2: Modify OSPF Priority and Force Elections

Scenario

In this activity, you will examine DR and BDR roles and watch the roles change when there is a change in the network. You will then modify the priority to control the roles and force a new election. Finally, you will verify routers are filling the desired roles.

Instructions

Part 1: Examine DR and BDR Changing Roles

Step 1: Wait until the amber link lights turn green.
When you first open the file in Packet Tracer, you may notice that the link lights for the switch are amber. These link lights will stay amber for 50 seconds while the STP protocol on the switch makes sure that one of the routers is not another switch. Alternatively, you can click **Fast Forward Time** to bypass this process.

Step 2: Verify the current OSPF neighbor states.
Use the appropriate command on each router to examine the current DR and BDR:

Router# show ip ospf neighbor

Step 3: Turn on IP OSPF adjacency debugging.
On RA and RB:

RA# debug ip ospf adj

RB# debug ip ospf adj

Step 4: Disable the Gigabit Ethernet 0/0 interface on RC.

RC(config)# interface g0/0

RC(config-if)# shutdown

Wait about 30 seconds for the dead timers to expire on RA and RB.

Step 5: Restore the Gigabit Ethernet 0/0 interface on RC.

RC(config)# interface g0/0

RC(config-if)# no shutdown

Wait for the new DR/BDR elections to occur.

Step 6: Disable the Gigabit Ethernet 0/0 interface on RB.
Disable the link to trigger another election and wait for timers to expire.

Step 7: Restore the Gigabit Ethernet 0/0 interface on RB.

RB(config)# interface g0/0

RB(config-if)# no shutdown

Step 8: Turn off Debugging.

RA# undebug all

RB# undebug all

Part 2: Modify OSPF Priority and Force Elections

Step 1: Configure OSPF priorities on each router.

On RA:

RA(config)# interface g0/0

RA(config-if)# ip ospf priority 200

On RB:

RB(config)# interface g0/0

RB(config-if)# ip ospf priority 100

On RC:

RC(config)# interface g0/0

RC(config-if)# ip ospf priority 1

Step 2: Force an election by resetting the OSPF process.

On each router, starting with RA:

Router# clear ip ospf process

Step 3: Verify DR and BDR elections were successful.

Use:

Router# show ip ospf neighbor

Wait for convergence or use **Fast Forward Time**.

2.6.6 Cisco Packet Tracer – Verify Single-Area OSPFv2

Objectives

- Identify and verify the status of OSPF neighbors.
- Determine how the routes are being learned in the network.
- Explain how the neighbor state is determined.
- Examine the settings for the OSPF process ID.
- Add a new LAN into an existing OSPF network and verify connectivity.

Background / Scenario

You are the network administrator for a branch office of a larger organization. Your branch is adding a new wireless network into an existing branch office LAN. The existing network is configured to exchange routes using OSPFv2 in a single-area configuration. Your task is to verify the operation of the existing OSPFv2 network before adding in the new LAN. When you are sure that the current OSPFv2 LAN is operating correctly, you will connect the new LAN and verify that OSPF routes are being propagated for the new LAN. As the branch office network administrator, you have full access to the IOS on routers R3 and R4. You only have read access to the enterprise LAN routers R1 and R2 using the username BranchAdmin and the password Branch1234.

Instructions

Part 1: Verify the Existing OSPFv2 Network Operation

The following commands will help:

- show ip interface brief
- show ip route
- show ip route ospf
- show ip ospf neighbor

- show ip protocols
- show ip ospf
- show ip ospf interface

Step 1: Verify OSPFv2 Operation on R1

- Log into R1 using BranchAdmin / Branch1234.
- Use show ip route.
- Use show ip ospf neighbor.
- Use the PC1 command prompt to ping the ISP router's address.

Step 2: Verify OSPFv2 Operation on R2

- Log into R2 using BranchAdmin / Branch1234.
- Use show ip route.
- Use show ip ospf interface g0/0.
- Use the PC2 command prompt to ping S0/0/1 on R3.

Step 3: Verify OSPFv2 Operation on R3

- Use show ip protocols.
- Use show ip ospf neighbor detail.
- Use PC3 to ping the ISP router's address.

Part 2: Add the New Branch Office LAN to the OSPFv2 Network

Step 1: Verify the OSPFv2 Configuration on Router R4

- Use show run | begin router ospf to verify network statements.

Step 2: Connect the Branch Office Router R4 to the OSPFv2 Network

- Use a **copper straight-through cable** to connect G0/0/0 on R4 to G0/1 on switch S3.
- Use show ip ospf neighbor on R4 and R3 to verify adjacency and OSPF state.
- Use the command prompt on the Laptop to ping PC2

4.1.4 Cisco Packet Tracer – ACL Demonstration

Objectives

- Part 1: Verify Local Connectivity and Test Access Control List
- Part 2: Remove Access Control List and Repeat Test

Background

In this activity, you will observe how an access control list (ACL) can be used to prevent a ping from reaching hosts on remote networks. After removing the ACL from the configuration, the pings will be successful.

Instructions

Part 1: Verify Local Connectivity and Test Access Control List

Step 1: Ping Devices on the Local Network to Verify Connectivity

- From the command prompt of PC1, ping PC2.
- From the command prompt of PC1, ping PC3.

Step 2: Ping Devices on Remote Networks to Test ACL Functionality

- From the command prompt of PC1, ping PC4.
- From the command prompt of PC1, ping the DNS Server.

Part 2: Remove the ACL and Repeat the Test

Step 1: Use Show Commands to Investigate the ACL Configuration

- Navigate to the R1 CLI.
- Use the show run and show access-lists commands to view the ACL configuration.
- Use show run | include interface|access to determine where the ACL is applied.

Step 2: Remove Access List 11 from the Configuration

a. Under the Serial0/0/0 interface, remove the ACL:

R1(config)# interface s0/0/0

R1(config-if)# no ip access-group 11 out

b. In global configuration mode, remove the ACL itself:

R1(config)# no access-list 11

c. Verify that PC1 can now ping the DNS Server and PC4.

5.1.8 Cisco Packet Tracer – Configure Numbered Standard IPv4 ACLs

Objectives

- **Part 1:** Plan an ACL Implementation
- **Part 2:** Configure, Apply, and Verify a Standard ACL

Background / Scenario

Standard access control lists (ACLs) are router configuration scripts that control whether a router permits or denies packets based on the source address. This activity focuses on defining filtering criteria, configuring standard ACLs, applying ACLs to router interfaces, and verifying and testing the ACL implementation. The routers are already configured, including IP addresses and Enhanced Interior Gateway Routing Protocol (EIGRP) routing.

Instructions

Part 1: Plan an ACL Implementation

Step 1: Investigate the Current Network Configuration

- Confirm full network connectivity by pinging between devices.

Step 2: Evaluate Two Network Policies and Plan ACL Implementations

- On **R2**:
  - Deny access from the 192.168.11.0/24 network to the WebServer at 192.168.20.254.
  - Permit all other access.
  - Place the ACL on the **outbound** interface to the WebServer.
- On **R3**:
  - Deny communication between the 192.168.10.0/24 network and the 192.168.30.0/24 network.
  - Permit all other access.

- Place the ACL on the **outbound** interface to PC3.

Part 2: Configure, Apply, and Verify a Standard ACL

Step 1: Configure and Apply a Numbered Standard ACL on R2

- Create ACL 1:

R2(config)# access-list 1 deny 192.168.11.0 0.0.0.255

R2(config)# access-list 1 permit any

- Verify the ACL:

R2# show access-lists

- Apply the ACL on interface GigabitEthernet0/0:

R2(config)# interface GigabitEthernet0/0

R2(config-if)# ip access-group 1 out

Step 2: Configure and Apply a Numbered Standard ACL on R3

- Create ACL 1:

R3(config)# access-list 1 deny 192.168.10.0 0.0.0.255

R3(config)# access-list 1 permit any

- Verify the ACL:

R3# show access-lists

- Apply the ACL on interface GigabitEthernet0/0:

R3(config)# interface GigabitEthernet0/0

R3(config-if)# ip access-group 1 out

Step 3: Verify ACL Configuration and Functionality

- Use show run or show ip interface gigabitethernet 0/0 to confirm ACL application.
- Perform pings across the network to confirm traffic is filtered as defined in Part 1.
- Use show access-lists to check if packets are matching the ACL rules.

16.3.1 Packet Tracer – Troubleshoot Static and Default Routes

Objectives

- Troubleshoot IPv4 static routes.
- Troubleshoot IPv4 default routes.
- Troubleshoot IPv6 static routes.
- Configure IPv4 static routes.
- Configure IPv4 default routes.
- Configure IPv6 static routes.

Background / Scenario

A newly hired network technician is attempting to preconfigure a simple topology that will be delivered to a customer. The technician has not been able to establish connectivity between the three LANs. You have been asked to troubleshoot the topology and verify connectivity between the hosts on the three LANs over IPv4 and IPv6.

Instructions

Step 1: Locate and Document the Problems

- Review the routing tables on routers R1, R2, and R3 using the following commands:
    - show ip route
    - show ipv6 route
- Analyze the configuration for incorrect or missing routes for both IPv4 and IPv6.

Step 2: Repair the Problems

- Configure the routers so that full connectivity exists between all hosts across the three LANs using the existing static route design.

- Do not change the static route types.
- Use the appropriate ip route and ipv6 route commands to correct any issues discovered in the routing configuration.

13.4.5 Packet Tracer – Troubleshoot WLAN Issues

Objectives

- Troubleshoot wireless LAN connectivity issues in a home network.
- Troubleshoot wireless LAN connectivity issues in an enterprise network.

Background / Scenario

Now that you have learned how to configure wireless in home and enterprise networks, you need to learn how to troubleshoot in both wireless environments. Your goal is to enable connectivity between hosts on the networks to the web server by both IP address and URL. Connectivity between the home and enterprise networks is not required.

To access the Home Wireless Router, the username and password is admin.

The WLC management interface username is admin and the password is Cisco123.

Instructions

Part 1: Troubleshoot the Network

Note: You will only be troubleshooting the Home Wireless Router, WLC, and wireless host devices in this activity.

Step 1: Test Connectivity

- Test connectivity between the various wireless hosts and the web server by both IP address and the URL www.netacad.pt.

Step 2: Investigate Issues

- Investigate the connectivity issues with each host. Issues may be with the host configuration or with other wireless network components.

Part 2: Fix Issues

- Make configuration changes to devices so that hosts can achieve connectivity with the network.
- Test to ensure all hosts can reach the communication goal of connecting to the web server by both IP address and URL.

15.6.1 Packet Tracer – Configure IPv4 and IPv6 Static and Default Routes

Objectives

- Configure IPv4 static and floating static default routes.
- Configure IPv6 static and floating static default routes.
- Configure IPv4 static and floating static routes to internal LANs.
- Configure IPv6 static and floating static routes to the internal LANs.
- Configure IPv4 host routes.
- Configure IPv6 host routes.

Background / Scenario

In this activity, you will configure IPv4 and IPv6 default static and floating static routes.

Note: The static routing approach used in this lab is intended to assess your ability to configure different types of static routes. It may not reflect networking best practices.

Instructions

Part 1: Configure IPv4 Static and Floating Static Default Routes

The network requires static routes to provide internet access to internal LAN users through the ISPs. Additionally, ISP routers require static routes to reach the internal LANs.

Step 1: Configure an IPv4 Static Default Route

- On Edge_Router, configure a directly connected IPv4 default static route via ISP1.

Step 2: Configure an IPv4 Floating Static Default Route

- On Edge_Router, configure a directly connected IPv4 floating static default route via ISP2 with administrative distance 5.

Part 2: Configure IPv6 Static and Floating Static Default Routes

Step 1: Configure an IPv6 Static Default Route

- On Edge_Router, configure a next-hop IPv6 static default route via ISP1.

Step 2: Configure an IPv6 Floating Static Default Route

- On Edge_Router, configure a next-hop IPv6 floating static default route via ISP2 with administrative distance 5.

Part 3: Configure IPv4 Static and Floating Static Routes to the Internal LANs

Step 1: Configure IPv4 Static Routes

- On ISP1, configure next-hop IPv4 static routes to LAN 1 and LAN 2 via Edge_Router.

Step 2: Configure IPv4 Floating Static Routes

- On ISP1, configure directly connected floating static routes to LAN 1 and LAN 2 via ISP2 with administrative distance 5.

Part 4: Configure IPv6 Static and Floating Static Routes to the Internal LANs

Step 1: Configure IPv6 Static Routes

- On ISP1, configure next-hop IPv6 static routes to LAN 1 and LAN 2 via Edge_Router.

Step 2: Configure IPv6 Floating Static Routes

- On ISP1, configure next-hop IPv6 floating static routes to LAN 1 and LAN 2 via ISP2 with administrative distance 5.

Part 5: Configure Host Routes

Corporate users frequently access a customer server. Static host routes will be configured for redundancy.

Step 1: Configure IPv4 Host Routes

- On Edge_Router, configure a directly connected host route to the customer server.
- Configure a floating static host route to the server with administrative distance 5.

Step 2: Configure IPv6 Host Routes

- On Edge_Router, configure a next-hop IPv6 host route to the customer server via ISP1.
- Configure a directly connected floating host route to the customer server via ISP2 with administrative distance 5.

14.3.5 Packet Tracer – Basic Router Configuration Review

Objectives

Part 1: Configure Devices and Verify Connectivity

- Assign static IPv4 and IPv6 addresses to the PC interfaces.
- Configure basic router settings.
- Configure the router for SSH.
- Verify network connectivity.

Part 2: Display Router Information

- Retrieve hardware and software information from the router.
- Interpret the startup configuration.
- Interpret the routing table.
- Verify the status of the interfaces.

Background / Scenario

This activity requires you to configure the R2 router using the settings from the Addressing Table and the specifications listed. The R1 router and the devices connected to it have been configured. This is a comprehensive review of previously covered IOS router commands. In Part 1, you will complete basic configurations and interface settings on the router. In Part 2, you will use SSH to connect to the router remotely and utilize IOS commands to retrieve information from the device.

Instructions

Part 1: Configure Devices and Verify Connectivity

Step 1: Configure the PC Interfaces

- Configure IPv4 and IPv6 addresses on PC3 and PC4 based on the addressing table.

Step 2: Configure the Router

- Set hostname.
- Configure encrypted privileged EXEC password.
- Set domain name.
- Disable DNS lookup.
- Enable password encryption.
- Create SSH user credentials.
- Generate RSA keys.
- Configure console line settings (password, exec-timeout, logging synchronous, login).
- Configure VTY line settings (SSH only, password, exec-timeout, login local).
- Set banner MOTD.
- Enable IPv6 unicast routing.
- Configure IPv4/IPv6 addresses and descriptions on all interfaces and enable them.
- Save the running configuration.

Step 3: Verify Network Connectivity

- Ping between PC3 and PC4 using IPv4 and IPv6.
- From R2, ping S0/1/1 on R1 using both protocols.
- From PC3, ping the ISP address and test external connectivity.
- From PC3, initiate SSH session to R2.

Part 2: Display Router Information

Step 1: Establish SSH Session to R2

- SSH into the router using the configured IPv6 address and credentials.

Step 2: Retrieve Hardware and Software Information

- Use show commands (e.g., show version) to collect router system information.

Step 3: Display the Running Configuration

- Use show running-config with filters to analyze password configuration and VTY settings.

Step 4: Display the Routing Table

- Use show ip route to view routing table entries and interface statuses.

Step 5: Display a Summary of Interfaces

- Use show ip interface brief and show ipv6 interface brief to verify addressing and interface status.

13.5.1 Packet Tracer – WLAN Configuration

Objectives

- Configure a home router to provide Wi-Fi connectivity to a variety of devices.
- Configure WPA2-PSK security on a home router.
- Configure interfaces on a WLC.
- Configure WLANs on a WLC.
- Configure WPA2-PSK security on a WLAN and connect hosts to WLAN.
- Configure WPA2-Enterprise on a WLAN and connect hosts to the WLAN.
- Verify WLAN connectivity.

Background / Scenario

You will apply your WLAN skills and knowledge by configuring a home wireless router and an enterprise WLC. You will implement both WPA2-PSK and WPA2-Enterprise security. Finally, you will connect hosts to each WLAN and verify connectivity.

Instructions

Part 1: Configure a Home Wireless Router

Step 1: Change DHCP Settings

- Open the Home Wireless Router GUI.
- Change the router IP and DHCP settings according to the Addressing Table.
- Set maximum DHCP addresses to 20.
- Set DHCP starting address to the third usable address in the LAN.
- Configure the internet interface for DHCP.
- Set the static DNS server based on the Addressing Table.

Step 2: Configure the Wireless LAN

- Configure the 2.4GHz wireless interface with the provided SSID.
- Use channel 6.
- Ensure the SSID is visible to all wireless hosts.

Step 3: Configure Security

- Configure WPA2-Personal using the passphrase from the WLAN table.
- Change the router's default administrative password.

Step 4: Connect Clients to the Network

- Connect the laptop, tablet, and smartphone to the wireless network.
- Verify connectivity among wireless hosts and to the web server.

Part 2: Configure a WLC Controller Network

Step 1: Configure VLAN Interfaces

- Access WLC-1 via web browser using Enterprise Admin PC.
- Configure interface **WLAN 2** with:
    - VLAN 2, Port 1
    - IP: 192.168.2.254
    - Gateway: RTR-1 G0/0/0.2
    - DHCP: same as gateway
- Configure interface **WLAN 5** with:
    - VLAN 5, Port 1
    - IP: 192.168.5.254
    - Gateway: RTR-1 G0/0/0.5
    - DHCP: same as gateway

Step 2: Configure a DHCP Scope

- Scope Name: management
- Pool Start: 192.168.100.235

- Pool End: 192.168.100.245
- Network: 192.168.100.0/24
- Default Router: 192.168.100.1

Step 3: Configure the WLC with External Server Addresses

- Add RADIUS server:
    - Index 1, IP: 10.6.0.254, Secret: RadiusPW
- Configure SNMP:
    - Community: WLAN
    - IP: 10.6.0.254

Step 4: Create the WLANs

- **WLAN 2**:
    - Profile: Wireless VLAN 2
    - SSID: SSID-2
    - ID: 2
    - Interface: WLAN 2
    - Security: WPA2-PSK (Passphrase: Cisco123)
    - Enable FlexConnect Local Switching & Auth
- **WLAN 5**:
    - Profile: Wireless VLAN 5
    - SSID: SSID-5
    - ID: 5
    - Interface: WLAN 5
    - Security: WPA2-Enterprise using RADIUS
    - Enable FlexConnect Local Switching & Auth

Step 5: Configure the Hosts

- Wireless Host 1: connect to Wireless VLAN 2

- Wireless Host 2: connect to Wireless VLAN 5 with credentials from WLAN info table

Step 6: Test Connectivity

- Verify that all wireless hosts can ping the web server and reach it via URL.