Module 1 – SE - Overview of IT Industry:

LAB EXERCISES:

1. **Write a simple "Hello World" program in two different programming languages of your choice. Compare the structure and syntax.**

Answer:

PART 1: "Hello World" Program in Two Languages

1) Python – Hello World

```
print("Hello World")
```

2) JavaScript – Hello World

```
console.log("Hello World");
```

PART 2: Comparison of Structure and Syntax

1) Syntax Simplicity

- **Python** has very simple and clean syntax.
- **JavaScript** syntax is slightly longer and uses semicolons (optional but common).

2) Keywords Used

- Python uses print() to display output.
- JavaScript uses console.log() to display output.

3) Code Structure

- Python code can run directly without any extra setup.
- JavaScript usually runs inside a browser or Node.js environment.

4) Line Endings

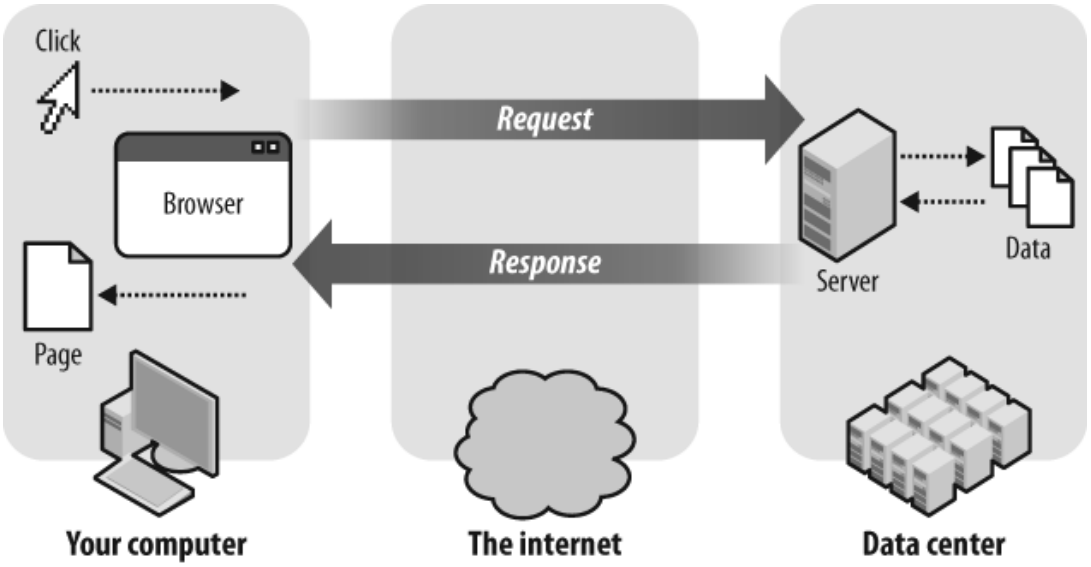
- Python does **not require semicolons**.
- JavaScript **commonly uses semicolons** at the end of statements.

5) Ease of Reading

- Python is considered **more readable and beginner-friendly**.
- JavaScript is slightly more complex but very powerful for web development.

2. **Research and create a diagram of how data is transmitted from a client to a server over the internet.**

Answer:



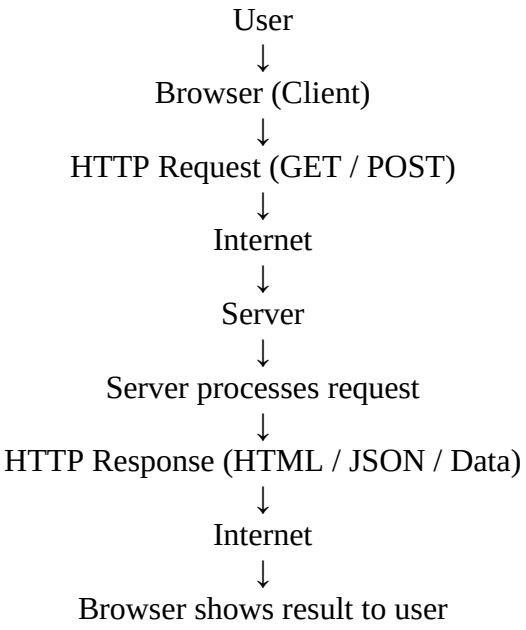
Explanation:

- **Client (your computer/browser)** sends a **request** (like opening a website).
- The request travels through **the internet**.
- **Server (in a data center)** receives the request, processes it, and fetches data.
- The server sends a **response** (web page/data) back to the client.
- The browser displays the page to the user.

Client asks → Server processes → Server replies → Client shows result

3. Design a simple HTTP client-server communication in any language.
Answer:

Simple Flowchart



Explanation:

- User performs an action (clicks a link or submits a form).
- The browser acts as a client and sends an HTTP request to the server.
- The request travels through the internet.
- The server receives the request and processes it.
- The server sends an HTTP response back.
- The browser receives the response and displays the result to the user.

4. Research different types of internet connections (e.g., broadband, fiber, satellite) and list their pros and cons.

Answer:

Different Types of Internet Connections

1. Broadband (DSL / Cable)

Pros:

- Widely available
- Affordable
- Suitable for daily use (browsing, streaming)

Cons:

- Speed depends on distance and users
- Slower than fiber
- Performance drops during peak hours

2. Fiber Internet

Pros:

- Very high speed
- Stable and reliable
- Best for gaming, streaming, and work

Cons:

- Limited availability
- Installation cost is high
- Not available in rural areas

3. Satellite Internet

Pros:

- Works in remote and rural areas
- No need for cables or towers

Cons:

- High latency (delay)
- Slower speed
- Affected by weather

4. Mobile Data (4G / 5G)

Pros:

- Portable and wireless
- Easy to set up
- Good speed (especially 5G)

Cons:

- Data limits
- Speed depends on signal strength
- Can be costly for heavy usage

Conclusion

- **Fiber** → Best performance
- **Broadband** → Budget-friendly
- **Satellite** → Remote areas
- **Mobile Data** → Portability

5. Simulate HTTP and FTP requests using command line tools (e.g., curl)

6. Identify and explain three common application security vulnerabilities. Suggest possible solutions

Answer:

1) SQL Injection

What it is:
SQL Injection happens when a hacker types special SQL code into input boxes (like login or search) to trick the database.

Example:
Entering
' OR '1'='1
in a login field can make the system think the login is always true, so it allows access without the correct password.

Solutions:

- Use **prepared statements** so user input is not treated as SQL code
- **Check and clean user input** before using it
- Never directly use user input in database queries

2) Cross-Site Scripting (XSS)

What it is:
XSS happens when a hacker inserts harmful JavaScript code into a website, and it runs in other users' browsers.

Example:
Adding a script in a comment box that steals user cookies or login data.

Solutions:

- **Check and filter user input**
- Convert special characters so scripts cannot run
- Use browser security rules like **Content Security Policy (CSP)**

3) Weak Authentication / Password Security

What it is:
Using simple or weak passwords makes accounts easy to hack.

Example:
Passwords like 123456, password, or admin.

Solutions:

- Force users to create **strong passwords**
- **Encrypt passwords** before saving them
- Use **two-step verification (OTP / 2FA)**

Summary:

- SQL Injection → Tricks the database
- XSS → Runs harmful scripts in browser
- Weak Passwords → Easy account hacking

7. **Identify and classify 5 applications you use daily as either system software or application software**

Answer:

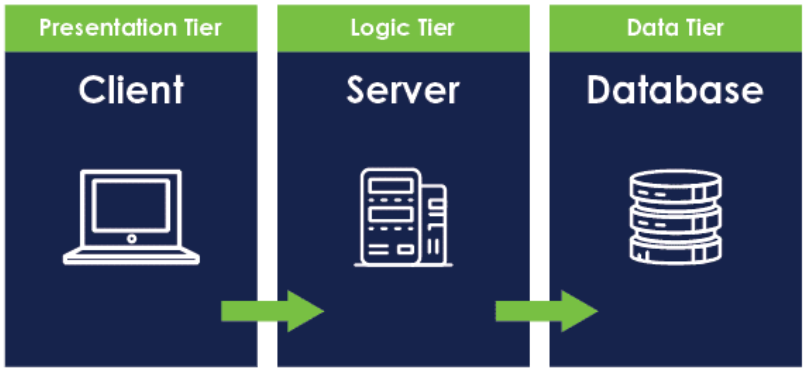
Identify and classify 5 applications used daily

Application	Type	Reason
Windows OS	System Software	Controls the computer hardware and allows other software to run
Google Chrome	Application Software	Used for browsing the internet
Microsoft Word	Application Software	Used to create and edit documents
WhatsApp	Application Software	Used for messaging and calling
VLC Media Player	Application Software	Used to play audio and video files

Explanation

- **System Software** helps run and manage the computer.
- **Application Software** helps users perform specific tasks.

8. **Design a basic three-tier software architecture diagram for a web application.**



This image shows a **basic three-tier web application architecture**:

- **Presentation Tier (Client):** The user interface where users interact (browser/app).
- **Logic Tier (Server):** Handles business logic, processes requests, and controls data flow.
- **Data Tier (Database):** Stores and manages application data.

9. Create a case study on the functionality of the presentation, business logic, and data access layers of a given software system.

Answer:

Case Study: Three-Tier Architecture of an E-Commerce Website

1. Presentation Layer (Client / UI)

The presentation layer is what the user sees and interacts with.
In an e-commerce website, this includes web pages like login, product listing, cart, and checkout.
It is responsible for displaying data and collecting user input (clicks, forms, searches).
This layer does **not** contain business rules.

Example:
HTML, CSS, JavaScript pages showing products and prices.

2. Business Logic Layer (Application / Server)

This layer processes user requests and applies rules of the system.
It decides *what should happen* when a user performs an action.

Responsibilities:

- Validate login credentials
- Calculate total price and discounts
- Handle order processing

Example:
Server-side code (Node.js, Java, Python) that checks stock availability before placing an order.

3. Data Access Layer (Database)

This layer stores and retrieves data.
It interacts directly with the database and ensures data consistency and security.

Responsibilities:

- Store user details
- Fetch product data
- Save orders and payments

Example:
Database systems like MySQL, MongoDB, or PostgreSQL.

Conclusion:

The three-tier architecture separates concerns, making the system:

- Easier to maintain
- More secure
- Scalable and flexible

Each layer works independently but together forms a complete software system.

10. Explore different types of software environments (development, testing, production). Set up a basic environment in a virtual machine.

Answer:

Types of Software Environments

1. Development Environment

- Used by developers to **write and test code**
- Changes happen frequently
- Errors are allowed (for learning and fixing)

Example:
Your laptop with VS Code, Node.js, Python, etc.

2. Testing Environment

- Used to **test the application before release**
- Checks if everything works correctly
- Bugs are found and fixed here

Example:
A separate server or VM where testers run test cases.

3. Production Environment

- The **live environment** used by real users
- Must be stable and secure
- No direct testing or experiments here

Example:
A live website like Amazon or Instagram.

Basic Environment Setup in a Virtual Machine

A **Virtual Machine (VM)** acts like a computer inside your computer.

Basic steps:

1. Install **VirtualBox / VMware**
2. Create a **new virtual machine**
3. Install an OS (Linux/Windows) inside it
4. Install required software (browser, editor, server tools)
5. Use it as a **testing or development environment**

Why VM is useful

- Keeps your main system safe
- You can test without risk
- Same setup can be reused anytime

11. Write and upload your first source code file to Github.
Answer:

How to do it:

Step 1: Write a simple code file

- Create a small file on your computer
  Examples:
  - hello.html (HTML)
  - hello.py (Python)
  - hello.js (JavaScript)
- Write a very basic program like "Hello World"
- Save the file

Step 2: Create a GitHub account

- Go to **https://github.com**
- Sign up or log in
- This will be your online place to store code

Step 3: Create a new repository

- Click **New Repository**
- Give it a simple name (e.g., first-code)
- Choose **Public**
- Click **Create Repository**

A repository is just a **folder for your code**.

Step 4: Upload your file

- Click **Add file → Upload files**
- Select your code file from your computer
- Click **Commit changes**

Now your code is uploaded

How you can submit:

- Your **GitHub repository link**
- Example format:
  https://github.com/your-username/first-code

12. Create a Github repository and document how to commit and push code changes.
Answer:

Creating a GitHub Repository and Documenting How to Commit & Push Code

1. **Create a GitHub account**
   Sign up on GitHub using your email ID.
2. **Create a new repository**
   - Click on "New Repository"
   - Give it a name (example: first-project)
   - Choose public or private
   - Click "Create"
3. **Add your source code file**
   - Write a simple program on your computer
   - Upload the file to the repository using GitHub's upload option
4. **Commit the code**
   - A commit means saving your changes with a short message
   - Example: "Added first source code file"
   - This helps track what changes were made
5. **Push the code to GitHub**
   - Pushing means sending your local code to the GitHub repository
   - After pushing, your code is visible online in the repository

13. Create a student account on Github and collaborate on a small project

with a classmate

14. Create a list of software you use regularly and classify them into the following categories: system, application, and utility software.

Answer:

List of Software Used Regularly and Their Classification

1) System Software

(System software manages hardware and provides a platform for other software)

- **Windows OS** – Manages computer hardware and runs applications
- **Android OS** – Controls smartphone hardware and apps

2) Application Software

(Application software helps users perform specific tasks)

- **Google Chrome** – Web browsing
- **Microsoft Word** – Document creation
- **Visual Studio Code** – Code editing and development
- **Spotify** – Music streaming
- **WhatsApp** – Messaging and communication

3) Utility Software

(Utility software helps maintain, optimize, or protect the system)

- **Windows Defender** – Virus and threat protection
- **Disk Cleanup** – Removes unnecessary files
- **WinRAR** – File compression and extraction
- **CCleaner** – System cleaning and optimization

Conclusion:

System software runs the device, application software helps users do work, and utility software maintains and secures the system.

15. Follow a GIT tutorial to practice cloning, branching, and merging repositories.
Answer:

What this means:

I studied the basic concepts of Git using an online tutorial and learned how developers manage code using Git.

What I learned:

- **Cloning**
  Cloning means making a copy of an existing project from GitHub to my local system so I can work on it.
- **Branching**
  Branching means creating a separate version of the code to work on new features or changes without affecting the main code.
- **Merging**
  Merging means combining the changes from one branch back into the main branch after the work is completed.

Outcome:
This practice helped me understand how Git is used for version control and how multiple developers can work together on the same project safely.

16. Write a report on the various types of application software and how they improve productivity
Answer:

Introduction

Application software is software designed to help users perform specific tasks such as writing documents, browsing the internet, communication, design, and data management. These applications make work faster, easier, and more organized, thus improving productivity.

Types of Application Software

1. Office Productivity Software

Examples: Microsoft Word, Excel, PowerPoint, Google Docs

Use:

- Creating documents
- Making spreadsheets
- Preparing presentations

How it improves productivity:

- Saves time compared to manual work
- Easy editing, formatting, and sharing
- Helps in professional documentation and reports

2. Communication Software

Examples: Gmail, WhatsApp, Microsoft Teams, Zoom

Use:

- Sending emails and messages
- Online meetings and video calls

How it improves productivity:

- Fast communication
- Reduces need for physical meetings
- Enables teamwork from different locations

3. Internet & Browsing Software

Examples: Google Chrome, Mozilla Firefox, Edge

Use:

- Browsing websites
- Searching information

How it improves productivity:

- Quick access to information
- Online learning and research
- Easy use of web-based tools

4. Multimedia Software

Examples: VLC Media Player, Spotify, YouTube

Use:

- Playing audio and video
- Learning through tutorials

How it improves productivity:

- Helps in learning visually and audibly
- Useful for relaxation and stress reduction

5. Design & Creative Software

Examples: Canva, Photoshop, Figma

Use:

- Designing posters, graphics, and UI

How it improves productivity:

- Quick design creation
- Useful for marketing and presentations
- Saves effort compared to manual designing

6. Educational Software

Examples: Coursera, Udemy, Google Classroom

Use:

- Online courses
- Learning new skills

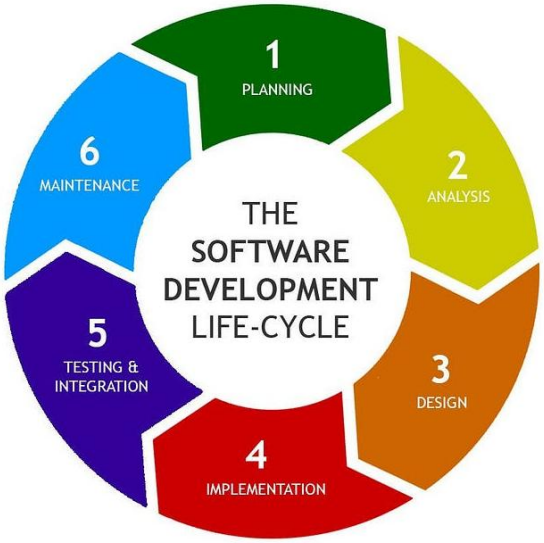
How it improves productivity:

- Skill improvement
- Learning anytime, anywhere
- Helps in career growth

Conclusion

Application software plays a very important role in daily life. It helps users complete tasks efficiently, saves time, improves accuracy, and increases overall productivity. Different types of application software are used for different purposes, but all aim to make work easier and faster.

17. Create a flowchart representing the Software Development Life Cycle (SDLC).
Answer:



- **Planning** → Decide what to build
- **Analysis** → Understand requirements
- **Design** → Plan system structure
- **Implementation** → Write code
- **Testing & Integration** → Find and fix bugs
- **Maintenance** → Update and improve software

The circular flow shows that SDLC is **continuous**, not one-time.

18. Write a requirement specification for a simple library management system.
Answer:

Requirement Specification: Library Management System

1. Introduction

The Library Management System is a simple software application used to manage library activities such as storing book details, managing members, issuing books, and returning books. It helps reduce manual work and helps librarians manage records easily.

2. Purpose

The purpose of this system is to:

- Maintain book records
- Manage library members
- Track book issue and return
- Make library operations simple and organized

3. Scope

- This system will be used by a small library.
- It will handle basic operations only and does not include advanced features like online payments or mobile apps.

4. Users of the System

- **Librarian** – manages books and members
- **Member/Student** – borrows and returns books

5. Functional Requirements

5.1 Book Management

- Add new books to the system
- Update book details
- Delete books if not needed
- Search books by title, author, or ID

5.2 Member Management

- Add new library members
- Update member information
- Delete member records
- View member details

5.3 Book Issue and Return

- Issue books to members
- Record issue date
- Return books
- Update book availability status

5.4 Reports

- View list of all books
- View issued books
- View available books

6. Non-Functional Requirements

- The system should be easy to use
- The system should respond quickly
- Data should be stored securely
- Only authorized users can access the system

7. Hardware Requirements

- Computer/Laptop
- Minimum 4 GB RAM

8. Software Requirements

- Operating System: Windows / Linux
- Browser: Chrome / Firefox
- Database: MySQL (or any simple database)

9. Constraints

- Internet connection may be required
- Only basic library features are included

10. Conclusion

The Library Management System will help manage library operations efficiently, reduce manual errors, and save time for librarians.

19. Perform a functional analysis for an online shopping system.

Answer:

Functional Analysis of an Online Shopping System

Functional analysis explains **what the system does** (features/functions), not how it is coded.

1. User Management

- User can **register** (create an account)
- User can **login / logout**
- User can **update profile details**
- User can **reset password**

2. Product Browsing

- User can **view product list**
- User can **search products**
- User can **filter products** (price, category, brand)
- User can **view product details** (price, description, images)

3. Shopping Cart

- User can **add products to cart**
- User can **remove products from cart**
- User can **update quantity**
- System shows **total price**

4. Order Processing

- User can **place an order**
- User can **select delivery address**
- User can **choose payment method**
- System generates **order confirmation**

5. Payment System

- Supports **online payment** (UPI, card, wallet)
- Supports **cash on delivery**
- Verifies **payment status**
- Generates **payment receipt**

6. Order Management

- User can **view order history**
- User can **track order status**
- User can **cancel order** (if allowed)
- Admin can **update order status**

7. Admin Functions

- Admin can **add/update/delete products**
- Admin can **manage users**
- Admin can **manage orders**
- Admin can **view sales reports**

8. Notifications

- Order confirmation message/email
- Payment success/failure notification
- Delivery updates

Conclusion

The online shopping system allows users to **browse products, place orders, make payments, and track deliveries**, while admins manage products, users, and orders efficiently.

20. Design a basic system architecture for a food delivery app.

21. Develop test cases for a simple calculator program.
Answer:

Test Cases for Simple Calculator

Test Case ID	Operation	Input 1	Input 2	Expected Output	Actual Output	Status
TC-01	Addition	5	3	8	—	—
TC-02	Addition	-2	4	2	—	—
TC-03	Subtraction	10	4	6	—	—
TC-04	Subtraction	5	8	-3	—	—
TC-05	Multiplication	6	7	42	—	—
TC-06	Multiplication	0	9	0	—	—
TC-07	Division	20	4	5	—	—
TC-08	Division	5	2	2.5	—	—
TC-09	Division	5	0	Error / Not Allowed	—	—
TC-10	Invalid Input	a	5	Error Message	—	—

22. Document a real-world case where a software application required critical maintenance
Answer:

Case Study: Critical Maintenance in WhatsApp (Facebook / Meta)

Software Application - WhatsApp Messenger

Problem Faced

In **2020**, WhatsApp faced a **critical security vulnerability** where hackers could crash the app or access user data by sending a specially crafted message. This affected millions of users worldwide.

Why Maintenance Was Critical

- User privacy and data were at risk
- App crashes reduced trust
- Large user base meant high impact
- Security flaw could be exploited remotely

Maintenance Action Taken

- WhatsApp developers identified the bug
- A **security patch update** was released quickly
- Users were advised to update the app immediately
- Older vulnerable versions were blocked

Outcome

- Security issue was fixed
- User data remained protected
- App stability improved
- Trust of users was maintained

Lesson Learned

- Regular updates are necessary
- Security testing is critical
- Fast maintenance prevents major damage

Conclusion:

This case shows how **timely software maintenance** is essential to ensure **security, reliability, and user trust** in real-world applications.

23. Create a DFD for a hospital management system.
24. Build a simple desktop calculator application using a GUI library.

25. Draw a flowchart representing the logic of a basic online registration system.
Answer:

Flowchart: Online Registration System

START
↓
Open Registration Page

↓
User Enters Details
(Name, Email, Password, etc.)
↓
Are all fields filled?
├── NO → Show error message → Go back to form
└── YES
↓
Is data valid?
(email format, password rules)
├── NO → Show validation error → Go back to form
└── YES
↓
Check if user already exists
├── YES → Show "User already registered" → END
└── NO
↓
Save user data in database
↓
Show "Registration Successful"
↓
END

How to explain this flowchart:

- User opens the registration page
- User fills the form
- System checks:
    - Are all fields filled?
    - Is the data valid?
    - Is the user already registered?
- If everything is correct → data is saved
- Success message is shown