Web Service and Service Oriented Programming – Revision Notes

☑ **Web Services – Definitions (As per different organizations)**

- **W3C**: Software application identified by a URI, with interfaces/bindings defined and discovered as **XML artifacts**.
- **Microsoft**: Programmable application logic accessible using **Standard Internet Protocols**.
- **IBM**: Interface exposing operations via **standardized XML messaging**.
- **SUN**: Software components **discovered, combined, and recombined** to solve user needs.

☑ **Web Services – Key Features**

- Built using **Object-Oriented Programming** principles.
- Operate over **open, standard Internet protocols** (HTTP, SMTP).
- Use **text-based (XML)** communication → easier debugging.
- Can **co-exist with or replace** traditional **EDI**.
- Facilitate **interoperability** across platforms/languages.

☑ **History Timeline**

1. Structured Programming
2. Object-Oriented Programming
3. Distributed Computing
4. EDI (Electronic Data Interchange)
5. World Wide Web
6. Web Services

☑ **EDI (Electronic Data Interchange)**

- Standardized **computer-to-computer** data exchange.
- Benefits:
  - **Lower costs**
  - **More accuracy**
  - **Faster trading cycle**
  - **Improved productivity**

☑ **Web Services Solve These Problems**

1. **Interoperability**: Cross-platform communication (unlike DCOM/RMI).
2. **Firewall Traversal**: Uses **HTTP (port 80)** — firewall-friendly.
3. **Complexity**: Simple, incremental implementation via open standards.

☑ **Web Service Characteristics**

- Communicates via **open protocols** (HTTP, SMTP).
- Processes **XML messages (SOAP)**.
- Describes services via **WSDL**.
- Discoverable via **UDDI**.

☑ **Service-Oriented Architecture (SOA)**

Entities:

- **Service Provider** – Offers the service.
- **Service Requestor** – Needs & invokes the service.
- **Service Broker** – Registry (searchable) of service descriptions.

Operations:

- **Publish** – Provider publishes WSDL to registry.
- **Find** – Requestor searches for required service.
- **Bind** – Requestor connects to provider and invokes the service.

☑ **Web Service Components**

Component	Purpose
XML	Data format – cross-platform, structured
SOAP	Messaging protocol (uses XML)
WSDL	Describes the service (contract)
UDDI	Registry to discover/publish services

☑ **XML (eXtensible Markup Language)**

- **Describes data**, not presentation.
- Tags are **user-defined**.
- Used for **cross-platform communication**.
- Can validate structure using **DTD/Schema**.

☑ **SOAP (Simple Object Access Protocol)**

- XML-based message protocol.
- Platform & language **independent**.
- Primarily uses **HTTP** (but can use SMTP/TCP).
- **Stateless** by default.
- Supports **RPC-style** or **document-style** messaging.

SOAP Message Structure:

- **Envelope** (required)
- **Header** (optional)
- **Body** (required)
- **Fault** (optional)

SOAP Characteristics:

- **Extensibility** – Can add features like security.
- **Neutrality** – Works over many transport protocols.
- **Independence** – Any programming model.

☑ **WSDL (Web Services Description Language)**

- **XML vocabulary** to describe web services.
- Specifies:
    - **Port Type** – Operations
    - **Message** – Input/Output
    - **Types** – Data used
    - **Binding** – Protocols and message formats

☑ **UDDI (Universal Description, Discovery, and Integration)**

- XML-based **registry framework**.
- Roles:
    - **Service Registry** – like yellow pages.
    - **Service Provider** – publishes services.
    - **Service Requestor** – finds and binds to services.

UDDI Benefits:

- Easier **discovery of services**.
- Broader **market reach**.
- Enables **automated e-commerce**.

Exam-Focused Quick Recall (MCQ Triggers)

- **SOAP** uses: **HTTP, XML**, supports **RPC** & **document style**.
- **WSDL**: Defines **what, how, where** of a web service.
- **UDDI**: Used for **service discovery**.
- **SOA Roles**: Provider, Requestor, Broker.
- **Envelope** is a **mandatory part** of SOAP.
- **XML**: Separates **data from presentation**.
- **Web services** solve: Interoperability, Firewall, Complexity.
- **Web service communication** is: **Text-based**, open protocols.