

**Institute of Computer Science  
Vikram University, Ujjain (M.P.)**

**SOFTWARE ENGINEERING**

**M.Sc. Computer Science**

**BY**

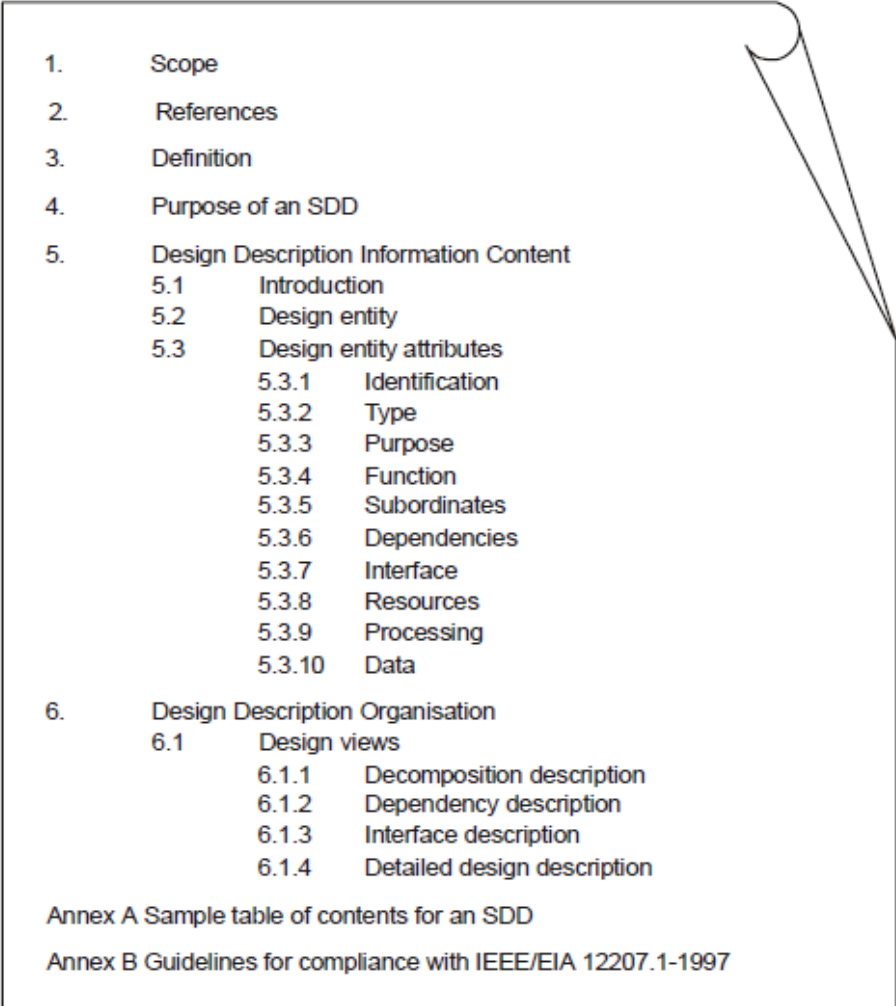
**Mr. Shekhar Disawal**

## SOFTWARE DESIGN DOCUMENTATION (SDD)

IEEE defines software design documentation as “*a description of software created to facilitate analysis, planning, implementation, and decision making. This design description is used as a medium for communicating software design information, and can be considered as a blueprint or model of the system*”.

While developing SDD, engineers should describe the design in sufficient detail that no further refinement of the tasks, databases, inter-task communications, libraries, module structure and interfaces, data structures, and databases is required in the code.

The information to be included in software design document depends on a number of factors, such as the type of software being developed and the approach used in its development. A number of standards have been suggested to develop software design document, however, the most widely used standard is by IEEE (see Figure Below), which acts as a general framework. This general framework can be customised and adapted to meet the needs of a particular organization.



1.	Scope
2.	References
3.	Definition
4.	Purpose of an SDD
5.	Design Description Information Content
5.1	Introduction
5.2	Design entity
5.3	Design entity attributes
5.3.1	Identification
5.3.2	Type
5.3.3	Purpose
5.3.4	Function
5.3.5	Subordinates
5.3.6	Dependencies
5.3.7	Interface
5.3.8	Resources
5.3.9	Processing
5.3.10	Data
6.	Design Description Organisation
6.1	Design views
6.1.1	Decomposition description
6.1.2	Dependency description
6.1.3	Interface description
6.1.4	Detailed design description

Annex A Sample table of contents for an SDD

Annex B Guidelines for compliance with IEEE/EIA 12207.1-1997

The SDD consist of several sections, which are discussed below:

- **Scope:** Identifies the release or version of the system being designed. The system is divided into modules; relationship between them and functionalities will be defined. Each iteration of the SDD document describes and identifies the software modules to be added or changed in a release.
- **References:** Lists references (both hardware and software documents and manuals) used in the creation of the SDD that may be of use to the designer, programmer, and user or to management personnel. This document is also considered useful for the readers of the document. In this section, any references made to the other documents including references to related project documents, especially the SRS is also listed. In addition, the existing software documentation (if any) is also listed.
- **Definition:** Provides a glossary of technical terms used in the document along with their definitions.
- **Purpose:** States the purpose of this document and its intended audience. This is primarily meant for the individuals who will be implementing the system.
- **Design description information content:** Consists of the sub-sections listed below:
  - **Introduction:** Since SDD is a representation of the design to be implemented, it should partition the system into design entities and describe the important properties and relationship among those entities.
  - **Design entity:** A design entity is a component of a design that is different in terms of structure and functions. The objective of creating design entities is to divide the system into separate components that can be considered, changed, and implemented. Each design entity possesses a unique name, purpose, and function but have common characteristics.
  - **Design entity attributes:** A design entity attribute is a property of the design entity, which provides factual information regarding the entity. Every attribute has an attached description, which includes references and design considerations. The attributes and their associated information are listed in Table below.

## Attributes and Description.

Attributes	Description
Identification	Identifies name of the entity. All the entities have a unique name.
Type	Describes the kind of entity. This specifies the nature of the entity.
Purpose	Specifies why the entity exists.
Function	Specifies what the entity does.
Subordinates	Identifies subordinate entity of an entity.
Dependencies	Describes relationships that exist between one entity and other entities.
Interface	Describes how entities interact among themselves.
Resources	Describes elements used by the entity that are external to the design.
Processing	Specifies rules used to achieve the specified functions.
Data	Identifies data elements that form part of the internal entity.

• **Design description organization:** Consists of the following sub-section:

□ **Design views:** Design views provide a comprehensive description of the design in a concise and usable form that simplifies information access and assimilation. There exist a number of ways to view the design where every design view represents a distinct concern about a system. The various design views and their attributes are listed in Table below.

### Design views and their description

Design View	Description	Attribute
Decomposition description	Partitions the system into design entities	Identification, type, purpose, function, and subordinate
Dependency description	Describes relationships between entities.	Identification, type, purpose, dependencies, and resources
Interface description	Consists of list that is required by the stakeholders (designers, developers and testers) in order to design entities.	Identification, function, and interfaces
Detail description	Describes internal details of the design entity.	Identification, processing, and data