

**Prasad V. Potluri Siddhartha Institute of Technology:: Vijayawada.
Department of Computer Science and Engineering**

I/II M.Tech. (CSE) (Second Semester)

17CSCS2T6B SOFTWARE ARCHITECTURE and DESIGN Credits: 4
Elective - IV

Lecture: 4 Periods/week

Internal Assessment: 40 Marks
Semester end examination: 60 Marks

Course Description:

Software Architecture and Design teaches the principles and concepts involved in the web analysis and design of large software systems.

Course Outcomes:

At the end of the course, the student is able:

CO1: Able to explain the process of decomposing large systems into physical hierarchies of smaller, more manageable components.

CO2: Able to identify the need for following good physical as well as logical design practices.

CO3: Learn specific techniques designed to eliminate cyclic, compile-time, and link-time dependencies.

CO4: Able to explain the top – down approach to the logical design of individual components

UNIT I

Introduction From C to C++

Using C++ to Develop Large Projects: Cyclic Dependencies, Excessive Link Time Dependencies, Excessive Compile time Dependencies, The global Name Space, and Logical Vs Physical Design Reuse, Quality: Quality Assurance, Quality Ensurance Software Development Tools. Preliminaries Multi-File C++ Programs: Declaration versus Definition, Internal versus External Linkage, Header Files, and Implementation (.c) Files Typedef Declarations, Assert Statements, A Few Matters of Style: Identifier Names, Class Member Layout, Iterators, Logical Design Notation: The IsA Relation, The Uses-In the Interface relation, the uses in the Implementation Relation Inheritance versus Layering, Minimality

UNIT II**Ground Rules:**

Overview, Member Data Access The Global Name Space: Global Data, Free functions, Typedefs and Constant data, preprocess of Macros. Names in header files Include Guards, Redundant Include Guards, Documentation, Identifier Naming Conventions.

UNIT III

Components.: Components versus Classes, Physical Design Rules, The Depends On Relation, Implied Dependency, Extracting Actual Dependencies, Friendship: Long Distance Friendship and Implied Dependency , Friendship and Fraud

UNIT III

Physical Hierarchy and Levelization: Physical Hierarchy: Acyclic Physical Dependencies, Level Numbers, Hierarchical and Incremental Testing, Cyclic Physical Dependencies, Cumulative Component Dependency (CCD)

Levelization: Some Causes of Cyclic Physical Dependencies: Enhancements, conveniences, Intrinsic Interdependency Escalation, Demotion,

UNIT IV

Insulation: From Encapsulation to Insulation: The cost of compile time coupling C++ Constructs and Compile-Time Coupling: Inheritance(IsA) and compile time coupling, Layering (hasA/ holdsA) and compile time coupling, inline functions and compile time coupling, private members compile time coupling , protected member compile time coupling , compiler generated member functions and compile time coupling , include directive and compile time coupling , default arguments and compile time coupling , enumerations and compile time coupling

TEXT BOOK:

1. Large-Scale C++ Software Design, John Lakos,1997, Addison-Wesley Professional

REFERENCE BOOKS:

1. Bjarne Stroustrup, “The C++ Programming Language”, Third Edition, Pearson Education