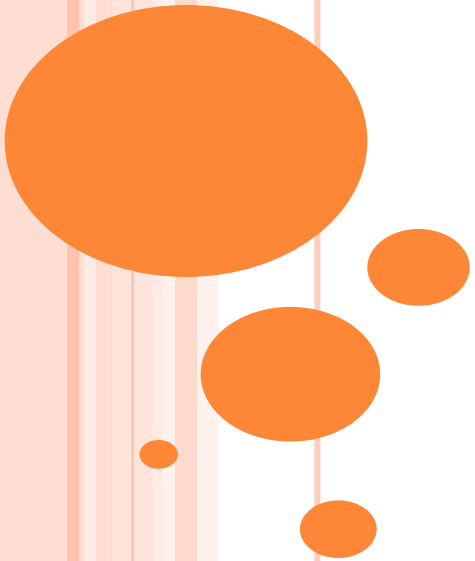
Bharathidasan University

Centre for Differently Abled Persons
Tiruchirappalli - 620024.

Programme Name	:	Bachelor of Computer Applications
Course Code	:	20UCA5CC7
Course Title	:	Software Engineering
Semester	:	V
Unit	:	Unit III
Compiled by	:	Dr. M. Prabavathy Associate Professor Ms.G.MAYA PRAKASH Guest Faculty

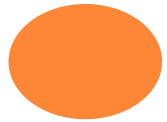


SOFTWARE DESIGN

SOFTWARE DESIGN

Software design is the process of investing and selecting programs that meet the objective for a software system.

- In order to be easily implementable in a conventional programming language, the following item must be designed during the design phase:
- Different modules required to implement the design solution.
- Control relationship among the identified modules.
- Interferes among different modules
- Algorithms required to implement the individual modules



Objectives of Software Design

Following are the purposes of Software design:

1. Correctness: Software design should be correct as per requirement.

2. Completeness: The design should have all components like data structures, modules, and

external interfaces, etc.

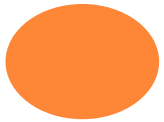
3. Efficiency: Resources should be used efficiently by the program.

4. Flexibility: Able to modify on changing needs.

5. Consistency: There should not be any inconsistency in the design.

6. Maintainability: The design should be so simple so that it can be easily maintainable by

other designers



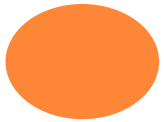
FUNDAMENTAL DESIGN CONCEPTS

1. Abstraction

- A solution is stated in large terms using the language of the problem environment at the highest-level abstraction.
- A collection of data that describes a data object is a data abstraction.

2. Architecture

- The complete structure of the software is known as software architecture.
- Structure provides conceptual integrity for a system in a number of ways.
- The architecture is the structure of program modules where they interact with each other in a specialized way.
- The components use the structure of data.



3. Patterns

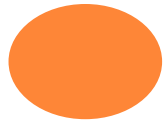
- A design pattern describes a design structure and that structure solves a particular design problem in a specified content.

4. Modularity

- A software is separately divided into name and addressable components.
- Modularity is the single attribute of a software that permits a program to be managed easily.

5. Information Hiding

- Modules must be specified and designed so that the information like algorithm and data presented in a module is not accessible for other modules not requiring that information.



6. Functional independence

- The functional independence is the concept of separation and related to the concept of modularity, abstraction and information hiding.

Cohesion

- Cohesion is an extension of the information hiding concept.
- A cohesive module performs a single task and it requires a small interaction with the other components in other parts of the program.

Coupling

- Coupling is an indication of interconnection between modules in a structure of software.

7. Refinement

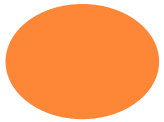
- Refinement is a top-down design approach.
- It is a process of elaboration.
- A program is established for refining levels of procedural details.

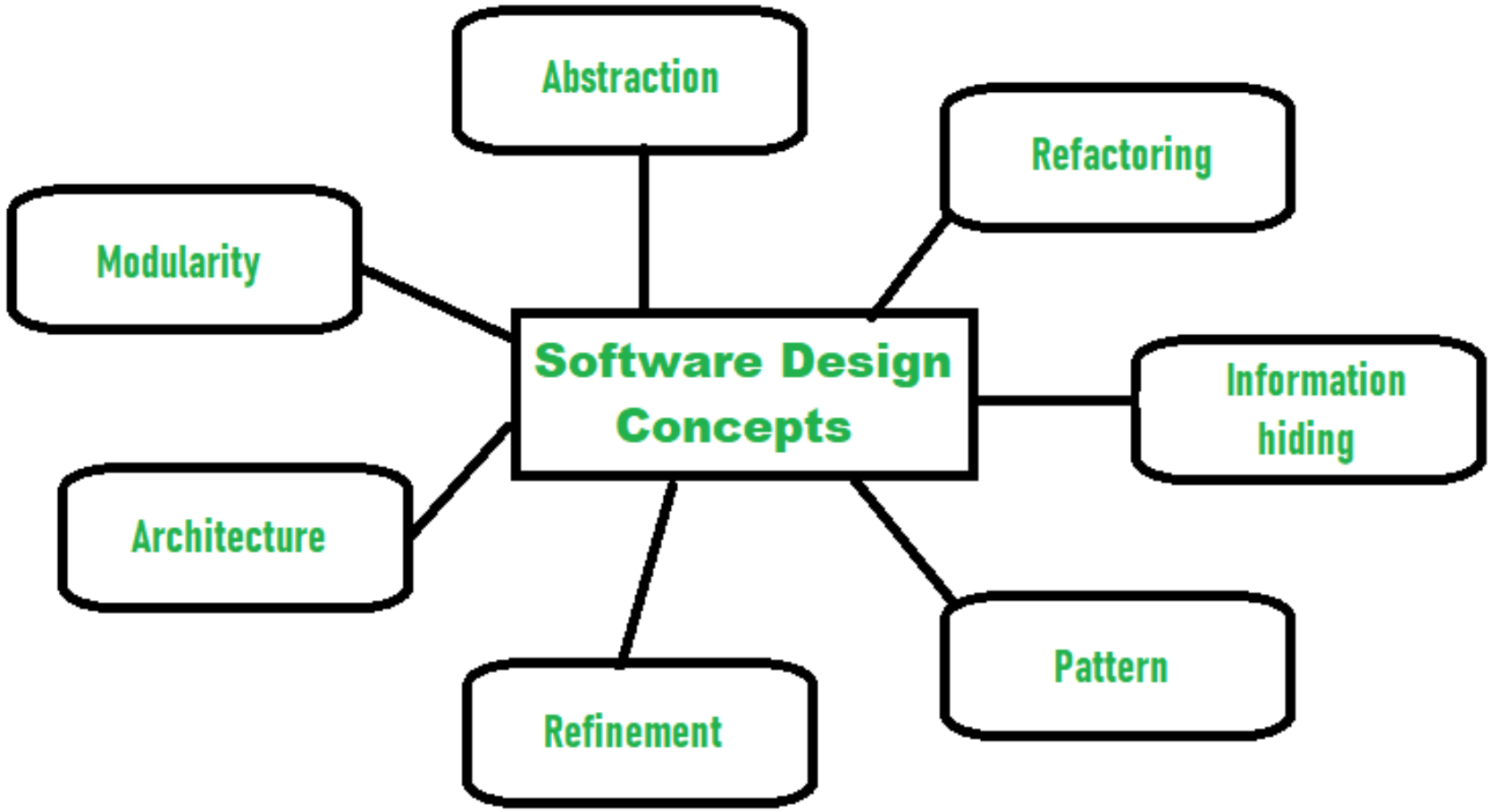
8. Refactoring

- Refactoring is the process of changing the software system in a way that it does not change the external behaviour of the code still improves its internal structure.

9. Design classes

- The model of software is defined as a set of design classes.




Abstraction
Refactoring
Modularity
Software Design Concepts
Information hiding
Architecture
Refinement
Pattern

SOFTWARE DESIGN MODELS

Software designs and problems are often complex and many aspects of software system must be modelled.

Software design models may be divided into 2 classes:

(i). Static Model

- A static model describes the static structure of the system being modeled, which is considered less likely to change than the functions of the system.

(ii). Dynamic Model

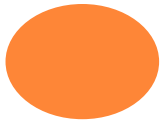
- Dynamic Modeling is used to represent the behavior of the static constituents of a software, here static constituents includes, classes, objects, their relationships and interfaces Dynamic

MODULARIZATION

- Modularization is a technique to divide a software system into multiple discrete and independent modules, which are expected to be capable of carrying out task(s) independently.

Advantage of modularization:

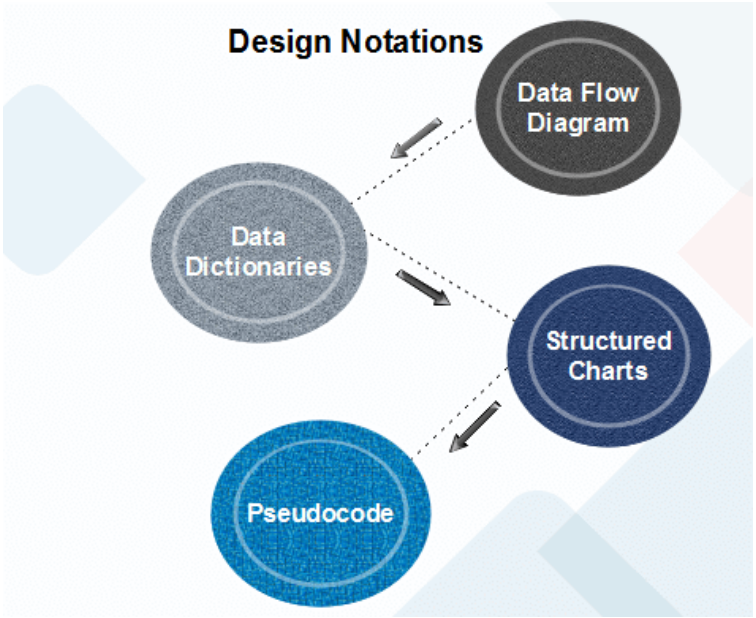
- Smaller components are easier to maintain
- Program can be divided based on functional aspects
- Desired level of abstraction can be brought in the program
- Components with high cohesion can be re-used again
- Concurrent execution can be made possible
- Desired from security aspect



DESIGN NOTATION

DESIGN NOTATION

Design Notations are primarily meant to be used during the process of design and are used to represent design or design decisions.

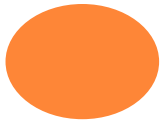


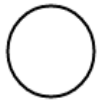
1. Data Flow Diagram

- Data-flow design is concerned with designing a series of functional transformations that convert system inputs into the required outputs.

- The design is described as data-flow diagrams.

- These diagrams show how data flows through a system and how the output is derived from the input through a series of functional transformations.



Symbol	Name	Meaning
	Rounded Rectangle	It represents functions which transforms input to output. The transformation name indicates its function.
	Rectangle	It represents data stores. Again, they should give a descriptive name.
	Circle	It represents user interactions with the system that provides input or receives output.
	Arrows	It shows the direction of data flow. Their name describes the data flowing along the path.
"and" and "or"	Keywords	The keywords "and" and "or". These have their usual meanings in boolean expressions. They are used to link data flows when more than one data flow may be input or output from a transformation.

2. Data Dictionaries

- A data dictionary lists all data elements appearing in the DFD model of a system.

- A data dictionary lists the objective of all data items and the definition of all composite data elements in terms of their component data items.

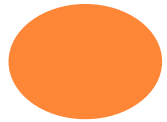
grossPay = regularPay + overtimePay

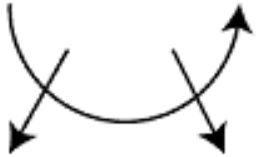
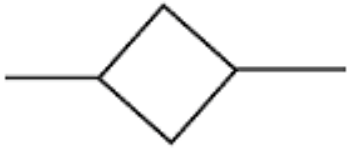
3. Structured Charts

A Black box system that functionality is known to the user without the knowledge of internal design.

Structured Chart is a graphical representation which shows:

- System partitions into modules
- Hierarchy of component modules
- The relation between processing modules
- Interaction between modules
- Information passed between modules



SYMBOL	DESCRIPTION
	Module
	Arrow
	Data couple
	Control Flag
	Loop
	Decision

REAL TIME SYSTEM

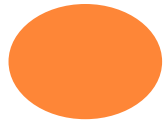
A real-time system is any information processing system which has to respond to externally generated input stimuli within a finite and specified period.

(i). Hard real-time systems

- An overrun in response time leads to potential loss of life and/or big financial damage
- Many of these systems are considered to be safety critical.

(ii). Soft real-time systems

- Deadline overruns are tolerable, but not desired.
- There are no catastrophic consequences of missing one or more deadlines.

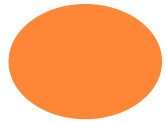


(iii) Firm teal-time systems

- The computation is obsolete if the job is not finished on time.
- Cost may be interpreted as loss of revenue.
- Typical example are forecast systems.

(iv). Weakly hard real-time systems

- Systems where m out of k deadlines have to be met.
- In most cases feedback control systems, in which the control becomes unstable with too many missed control cycles.
- Best suited if system has to deal with other failures as well (e.g. Electro Magnetic Interference EMI).

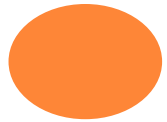


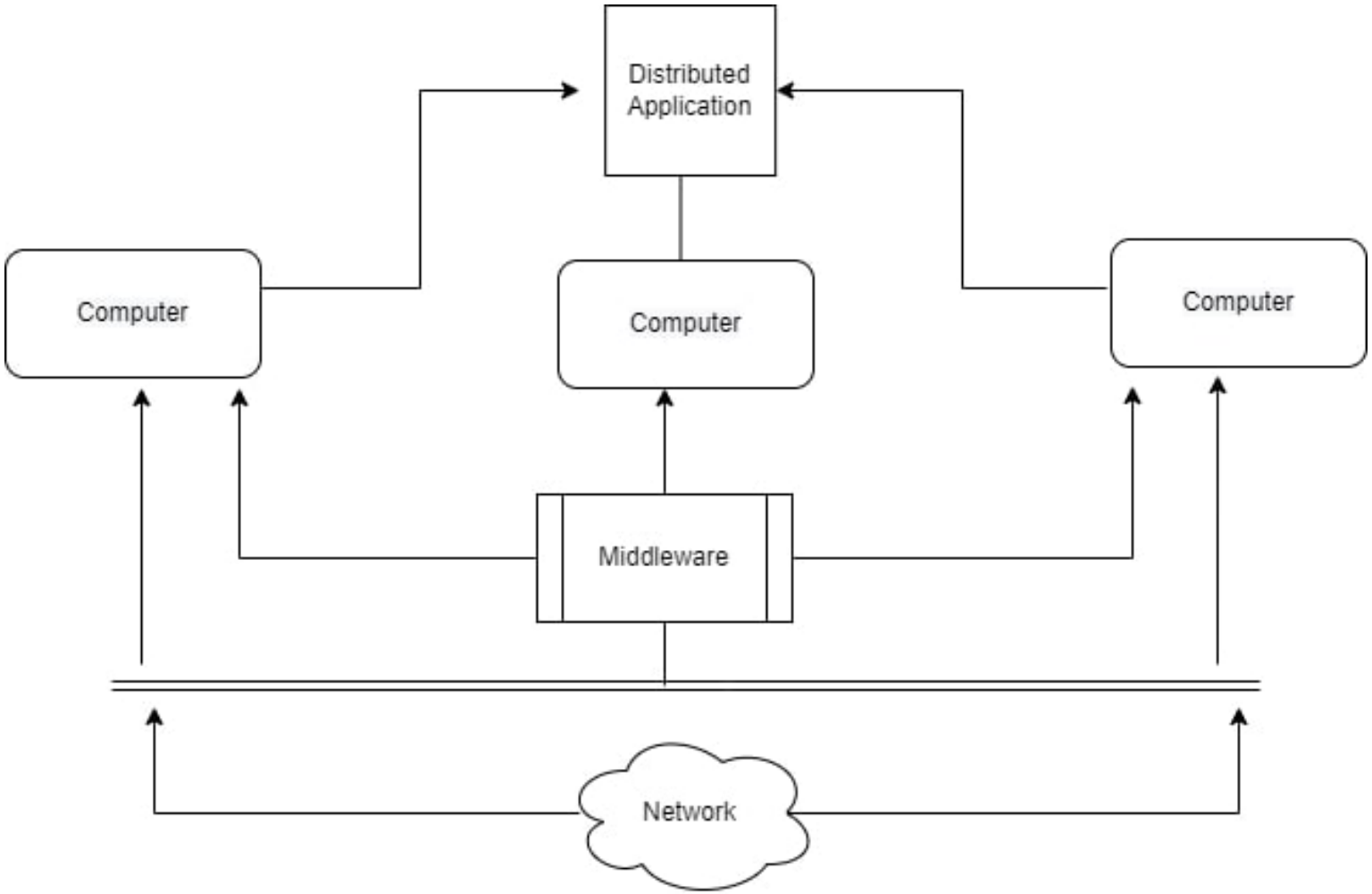
DISTRIBUTED SYSTEM

Distributed System is a collection of autonomous computer systems that are physically separated but are connected by a centralized computer network that is equipped with distributed system software.

Distributed System Software: This Software enables computers to coordinate their activities and to share the resources such as Hardware, Software, Data, etc.

Database: It is used to store the processed data that are processed by each Node/System of the Distributed systems that are connected to the Centralized network.




Distributed Application
Computer
Computer
Computer
Middleware
Network

THANK You