OBJECT ORIENTED PROGRAMMING USING JAVA (CST 205)

Prepared by ***Prof.Renetha J.B. LMCST***

MODULE 1

Introduction

Module 1 - Topics

- Approaches to Software Design
- Object Modeling Using Unified Modeling Language (UML)
- Introduction to Java

Module 1 - Topics

- **Approaches to Software Design**
- Object Modeling Using Unified Modeling Language (UML)
- Introduction to Java

Approaches to Software Design

√ Functional Oriented Design

√ Object Oriented Design

√ Case Study of Automated Fire Alarm System.

Software

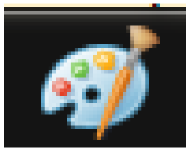
- Software
  - Software is a collection of instructions that *enable the user to interact with a computer*, its hardware or perform tasks.

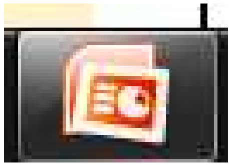
- Without software, most computers would be useless.
  - For example, without Internet browser software, we could not surf the Internet.
  - Without an operating system, the browser could not run on your computer.

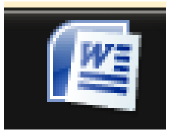
Software - types

- There are two types of software
  1. System Software – E.g- Operating System, Compilers, Interpreters etc

  1. Application Software - Microsoft Word, MS paint, Microsoft PowerPoint etc.

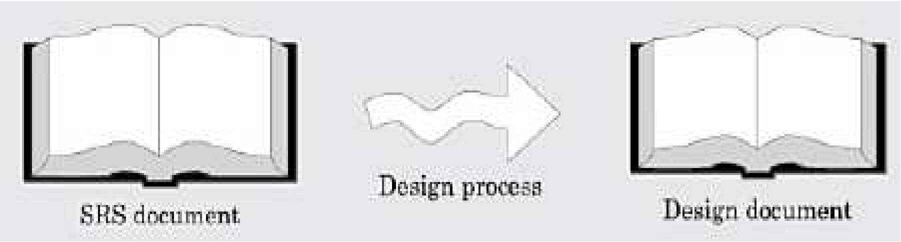






Software Design

- During the software design phase,
  - the design document is produced, based on the customer requirements (documented in the SRS - Software Requirements Specification document)



- The **design process** essentially *transforms the SRS document into a design document*.

Software Design

- The design document produced at the end of the design phase should be **implementable using a programming language** in the subsequent (coding) phase.

- Software design is the first step in SDLC (Software Design Life Cycle)

APPROACHES TO SOFTWARE DESIGN

There are two fundamentally different approaches to software design —

- Function-oriented design,
- Object-oriented design.

Functional Oriented Design (FOD)

- In function-oriented design, the system is comprised of many smaller sub-systems known as **functions**.

- These functions are capable of performing significant task in the system.

Functional Oriented Design (FOD)- Features

The salient features of the function-oriented design approach:

- Top-down decomposition
- Centralised system state

FOD-Top-down decomposition

- A system at starting level(high level) is viewed as a black box that provides certain services (also known as **high-level functions**) to the users of the system.
- This function may be refined into the **sub-functions**.
- Each of these sub-functions may be split into **more detailed subfunctions** and so on.

FOD-Top-down decomposition

- Function oriented design inherits some properties of structured design where divide and conquer methodology is used(**top-down approach**).

FOD (Top down design - example)

- Consider a Library management System
- Some of the functions in this system are:
  - *create-new-library member*
  - *issue book*
  - *return book*

FOD (Top down design - example)

- Consider one function **create-new-library-member** to creates the record for a new member,
- This function may be refined into the following subfunctions:
  - *assign-membership-number*
  - *create-member-record*
  - *print-bill*

Each of these sub-functions may be split into more detailed subfunctions and so on.

FOD-Centralised system state

- The system state is centralised and shared among different functions.
- The system state can be defined as the values of certain data items that determine the response of the system to a user action or external event.
- Such data in procedural programs usually have **global scope** and are **shared by many modules(functions)**.

FOD-Centralised system state(Eg.)

- For example, the set of books (i.e. whether borrowed by different users or available for issue) determines the state of a library automation system.

FOD-Centralised system state(Eg)

- For example, in the library management system, several functions ***share data*** such as *member-records* for reference and updation:
  - create-new-member
  - delete-member
  - update-member-record

Function-oriented design approaches

Some well-established function-oriented design approaches are :

- Structured design by Constantine and Yourdon
- Jackson's structured design by Jackson
- Warnier-Orr methodology
- Step-wise refinement by Wirth
- Hatley and Pirbhai's Methodology

Object oriented Design

- In the object oriented design approach, the system is viewed as **collection of objects** (i.e. entities).
- The state is **decentralized** among the objects and each object manages its own state information.- **no globally shared data**

Object oriented Design(contd)

- Each object is associated with a set of functions that are called its **methods**.
- Each object contains its own data and is responsible for managing it.
- The data internal to an object cannot be accessed directly by other objects
  - can be accessed only through invocation of the methods of the object.

Object oriented Design(contd)

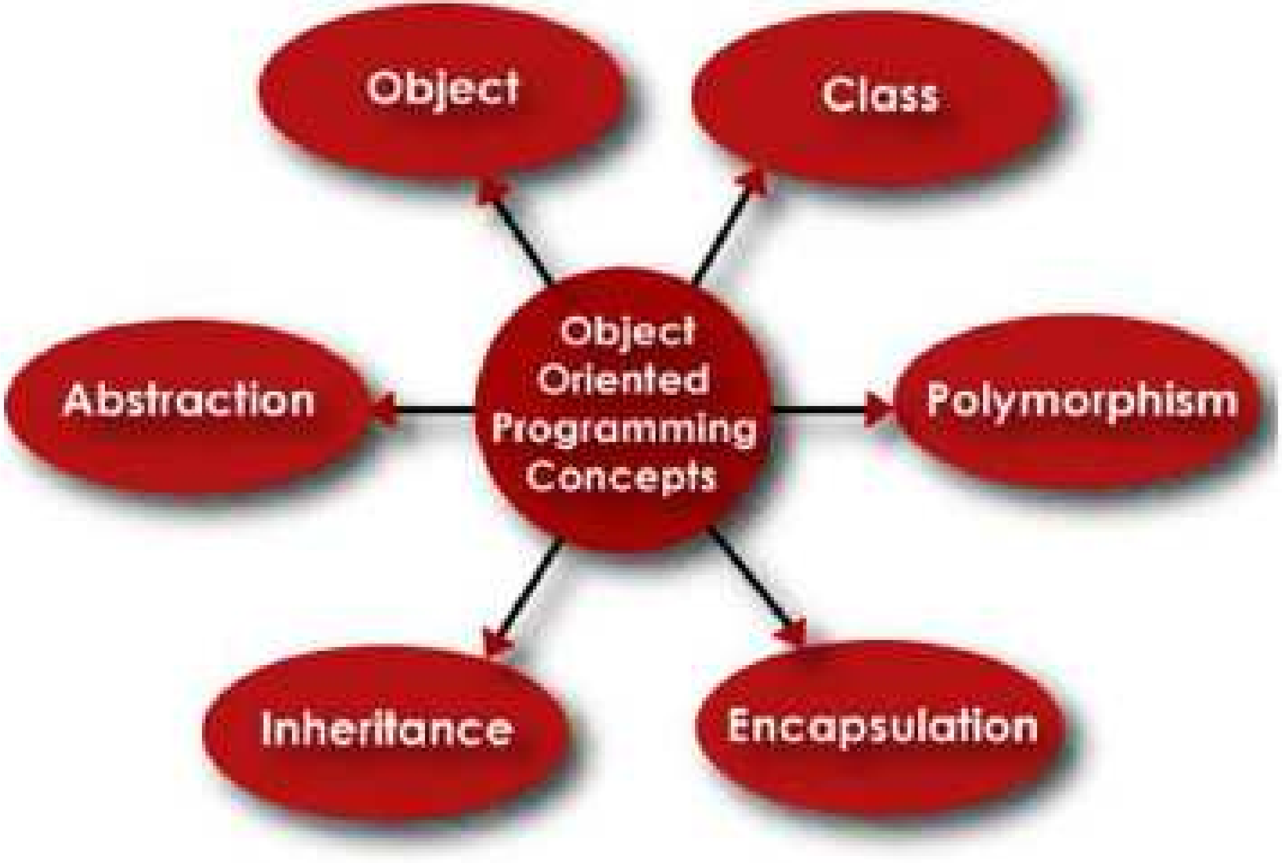
- The object-oriented design paradigm makes extensive use of the principles of **abstraction** and **decomposition.**
  - Objects decompose a system into functionally independent modules.
  - Objects can also be considered as instances of abstract data types (ADTs).

Object oriented Design(contd.)

Data abstraction(data hiding):

- The principle of data abstraction implies that how data is exactly stored is abstracted away.
- This means that any entity outside the object *will not have knowledge* about
  - how data is exactly stored, organised, and manipulated inside the object.
- The entities external to the object can access the data internal to an object only by calling certain well-defined methods supported by the object.

Basic Object Oriented concepts



Function Oriented Design

- The basic abstraction is the **services(functions)** that are available to the users of the system such as issue book ,display book details etc.
- state information is available in a **centralised shared data store**.
- Function oriented techniques group functions together (they constitute a higher level function)

Object Oriented design

- The basic abstraction is **real-world entities(objects)** such as member, book, book-register etc.
- state information exists in the form of **data distributed** among several objects of the system
- Object oriented techniques group functions together on the basis of the data they operate on.

Case Study of

Automated Fire Alarm System

Automated fire-alarm system—customer requirements

The owner of a large multi-storied building wants to have a computerised fire alarm system designed, developed, and installed in his building. Smoke detectors and fire alarms would be placed in each room of the building. The fire alarm system would monitor the status of these smoke detectors.

Automated fire-alarm system—customer requirements

The owner of a large multi-storied building wants to have a computerised fire alarm system designed, developed, and installed in his building. Smoke detectors and fire alarms would be placed in each room of the building.

The fire alarm system would monitor the **status of these smoke detectors.**

Automated fire-alarm system(contd.)

- Whenever a fire condition is reported by any of the smoke detectors, the fire alarm system **should determine the location** at which the fire has been sensed and then **sound the alarms only in the neighbouring locations.**
- The fire alarm system should also **flash an alarm message** on the computer console.

Automated fire-alarm system (contd.)

- Fire fighting personnel would man the console round the clock.
- After a fire condition has been successfully handled, the fire alarm system should support resetting the alarms by the fire fighting personnel.

Fire Alarm system-Function oriented design

```
/* Global data (system state) accessible by various functions */
BOOL detector_status [MAX_ROOMS];
int detector_locs [MAX_ROOMS];
BOOL alarm_status [MAX_ROOMS]; /* alarm activated when status is set */
int alarm_locs [MAX_ROOMS]; /* room number where alarm is located
int neighbor alarm[MAX_ROOMS][10]; /* each detector has at most 10 neighboring locations */
```

Fire Alarm system-Function oriented design

The **functions** which operate on the system state are:

interrogate_detectors()

get_detector_location()

determine_neighbor()

ring_alarm()

reset_alarm()

report_fire_location()

Fire Alarm system-Object oriented design

class detector

attributes: status, location, neighbors

operations:

create, sense_status, get_location, find_neighbors

class alarm

attributes: location, status

operations:

create, ring_alarm, get_location, reset_alarm

class sprinkler

attributes: location, status

operations: create, activate-sprinkler, get_location, reset-sprinkler

Analysis

Function oriented approach

- The system state (data) is **centralised** and several functions access and modify this central data.
- Data is **global** and can be easily accessed.
- The basic unit of designing a function oriented program is functions and modules.
- Functions appear as verbs.
- TOP DOWN

Object oriented approach

- The state information (data) is **distributed** among various objects.
- Data is **private** in different objects and cannot be accessed by the other objects.
- The basic unit of designing an object oriented program is objects.
- Objects appear as nouns.
- BOTTOM UP

Reference Text Book

- Rajib Mall, Fundamentals of Software Engineering, 4th edition, PHI, 2014.

THANK YOU