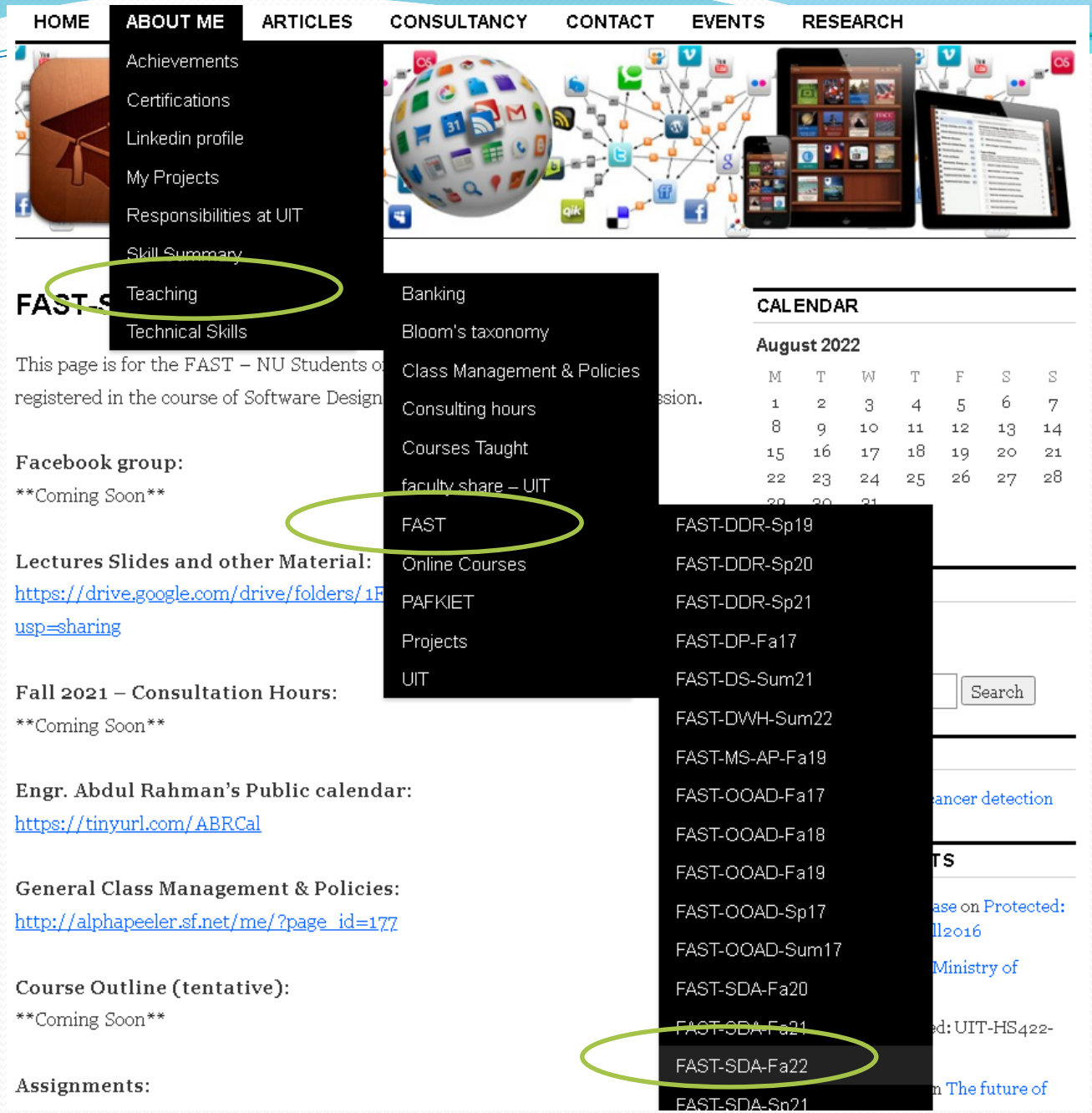
About The Course

Course portal: www.alphapeeler.sf.net/me/?page_id=665



Class Policies

alphapeeler.sourceforge.net/me/?page_id=177

Class Management & Policies

Engr. Abdul Rahman is expecting each student to follow Classroom / Lab Policies, & Procedures listed below:

A. Note from Engr. Abdul Rahman: I have established a few simple policies to lead a respectful and disciplined classroom. You are responsible to comply with the policies. If you fail to comply, there will be serious consequences.

B. Class / Lab Rules:

1. **Strict attendance policy**: Students are required to maintain 100% attendance throughout the session. 5 Minutes margin will be given after that student will be marked absent.
2. **No space for plagiarism**: Incase, if any of the assignment/project deliverables found plagiarized, the whole assignment/ project will be marked 'ZERO'.
3. **Late submission**: Within 1 day after deadline => 25% marks will be deducted. After 1 day => 50% marks will be deducted. After 2 days, 'ZERO' credit will be given.
4. **Submission of Assignment**: Students will submit their assignments within due date. If a student has an excused absence from class he or she is responsible for the assignments / homework that missed. It is up to the student to inquire about missed work and tests. Zero will be given if a student fails to make up work within an acceptable period. Following elements are mandatory for an assignment file:
    1. Assignment must be submitted in a proper file cover, and must be labeled properly.
    2. On cover page following items should be printed: Student name, Roll no, Date of submission.
    3. Attach print of the assignment question paper issued by the instructor after cover page.
    4. Attach hand written assignment after question paper.
5. **Consultation Time**: Students are advised to meet Engr. Abdul Rahman during the consultation time of the course only with prior appointment. Refer to the procedure for consulting hours from this url: http://alphapeeler.sourceforge.net/me/?page_id=158
6. **Project Submission**: The course required a proper project which will be submitted in Week 13. In this project, a proper report of at least 40 pages will be submitted after which a viva will be conducted in front of Engr. Abdul Rahman / HoD.
7. **Hand-held devices:** It is generally not acceptable to use cell phones, pagers, IPod/MP3 players, computers, etc. during lectures, except with the permission of Engr. Abdul Rahman and for reasons directly related to class activity.
8. **Lab assignments:** Assignments are checked only within lab timings. Lab files will not be entertained after lab timings.
9. **Courtesy and respect to all:** Students will exhibit courtesy and respect toward all other students at all times. Hateful comments concerning race, gender, sexuality, political views, appearance, or of any other type will not be tolerated; this applies to serious as well as "joking" comments.
10. **Leave the Food at Home:** Students may not eat in the classroom. This includes gum and candy. Drinks are also not permitted.
11. **Make-Up Tests:** There is no official policy defined for make-up tests, if you are absent or have not appeared in test then zero marks will be given to you.
12. **Final Year Students:** Students who are engaged in FYP, are responsible to demonstrate their work at least twice a week in FYP lab, otherwise I may send unsatisfactory report to the FYP coordinator.
13. **Leave policy:** Application of leave is not entertained by the class teacher, it should be notified to the HoD, and CC to Director Academics / Examination & Manager Student affairs. Even if the leave is approved, your class teacher will not mark you present on the basis of sick leave or any other type of leave. If you fail to maintain 75% attendance, you may not be eligible to sit in exams.

Class Policies

alphapeeler.sourceforge.net/me/?page_id=177

14. **Class compensation:** Engr. Abdul Rahman will notify the CR of the class in case of any class missed die to holidays or extra class required for students. It is the responsibility of class CR to schedule extra class by after reviewing the time table of class and teacher's time table and book the classroom from administration block.
15. **Late arrival application:** No application will be considered for late arrival after the class has been dismissed. Students need to submit their late arrival application on the same date during the class. Teacher has the right to dismiss the late arrival apology application in case of regular late arrivals.
16. **Entering the Classroom Procedure:** Enter the classroom quietly and in advance of class starting time. Class start time means that you are in your seat and working on your exercise. Class CR is responsible to turn on the multimedia projector before the class starts.
17. **Classroom Exit Procedure:** Wait for me to dismiss you.

C. Exam policies:

1. Read all questions carefully first and then ask for clarifications.
2. Question paper related queries will not be entertained after 30 minutes after start of paper.
3. Do not write anything on question paper unless until specifically asked for.
4. Fill the required information and return the question paper along with the answer script.
5. Write your name, and enrollment number, otherwise you may not remain eligible for exam.
6. Get your paper signed from invigilator against your enrollment number; else your paper will not be checked.
7. Only attempt questions assigned to your column, otherwise you may disqualify from exam.
8. In case of MCQs, only circle one choice, otherwise you may disqualify from exam.
9. Any kind of miss-conduct/miss-behavior/cheating will disqualify the candidate.
10. Warning will be issues only once, along with -1 score, after that you will lose your eligibility for exam.

D. If YOU CHOOSE to Break a Rule: Punishments will always fit the crime. Of course there are behaviors that will warrant a Vice Principal's Referral immediately. Examples of this include gross insubordination or violent behavior.
Behaviors that are less severe, but in violation of the basic rules of the class will be dealt with in the manner described below. This format is in no way all inclusive and is subject to change:
1st Incident — Teacher/Student Conference
2nd Incident — Teacher/Student Conference, Parent Notification by phone or email, review behavior grade per grading policy.
3rd Incident — Referral to Administration / discipline committee.
Note: All students are required to print a copy of this page and submit to the class teacher with their signatures in order to make sure that all rules are communicated to the students.

Reference books

- UML 2.0 in Action by Patrick Graessle
- Systems Analysis and Design with UML 3rd Edition by Alan Dennis
- *UML Distilled, 3rd Edition by Martin Flower*
- Visual Modeling with Rational Rose by Terry Quatranie
  Unified Software Development by Booch, Rambaugh, Jacobson

Assessment

- **Final Exam (50%)**
- **Home work assignments (10%)**
- **Term exams 30%**
- **Project(confirmed) 10%**

Course Goals

- After successful completion of this course students should be able to do analysis of software system. Do modeling using UML and create diagram like use cases, activity, class, sequence, collaboration etc.

Introduction to Systems Analysis and Design

What is Systems Analysis and Design?

- Systems are created to : solve problems.
- Think of the systems approach

  an organized way of dealing with a problem.
- System Analysis and Design, mainly deals with

  the software development activities.

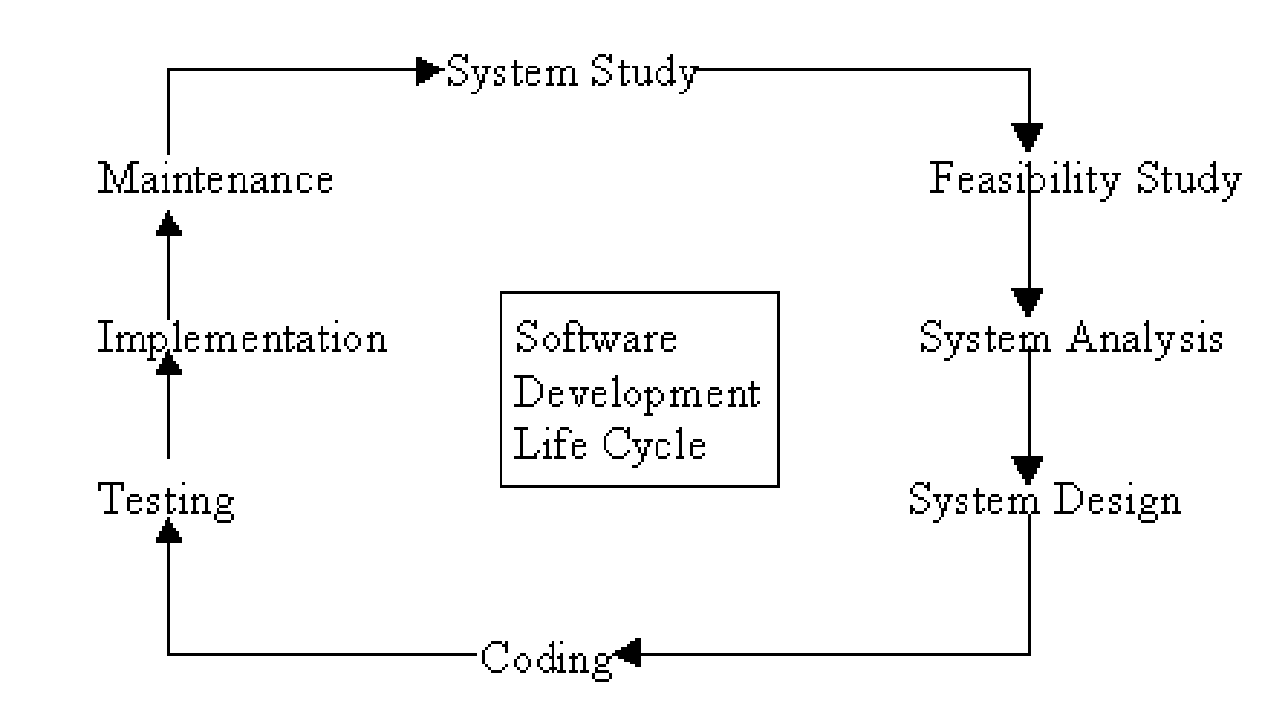
Defining A System

- A collection of components that work together to realize some objective forms a system.
- Basically there are three major components in every system, namely input, processing and output.

System Development Life Cycle

- System Development Life Cycle is an organizational process of developing and maintaining systems.
- System development life cycle means combination of various activities.
- Following are the different phases of software development cycle:
  - System study
  - Feasibility study
  - System analysis
  - System design
  - Coding
  - Testing
  - Implementation
  - Maintenance

System Development Life Cycle



System Environments ?

- Development
- Test
- Staging
- Pre-Production
- Production
- Mirror
- Roles involved:
  - D, DM, PM, TM, CM & DT.

System Study

- System study - **1st stage** of system development life cycle.
- Gives clear picture of what actually the physical system is.
- **System study phases (I & II):**
  - I: initial survey of the system - helps in identifying the scope.
  - II: in-depth study - requirement identification / limitations & issues of current system.
- **Proposal**:
  - Prepared after completing the system study,
  - prepared by the System Analyst.
  - Contains the findings of the current system
  - Recommendations to overcome the limitations / issues of the current system.
- **Steps of System study phase:**
  - problem identification and project initiation.
  - background analysis.
  - inference or findings.

Feasibility Study

- Done on the basis of ***initial study***.
- It is the test of the proposed system in the light of its workability, user's requirements, effective use of resources and the cost effectiveness.
- **Goal** : to achieve the scope.(not to solve issues)
- **Advantage**: Cost and benefits are estimated with greater accuracy.

System Analysis

- Analysis is detailed study of :
  - current system, leading to specifications of a new system.
  - system operations & its relationships within & outside system.
- **Data collection for**: files, decision points, & transactions of present system.
- **Tools of system analysis**: Interviews, on-site observation & questionnaire.
- **Steps to define boundary** of the new system:
  - Keeping in view the problems and new requirements
  - Workout pros & cons including new system
- **Analysis is documented in**: detailed DFDs, data dictionary, logical data structures & miniature specifications.
- Includes sub-dividing of complex process, data store identification & manual processes.

System Analysis Action items

- **Specification** of what the new system is to accomplish based on the user requirements.
- **Functional hierarchy** showing the functions to be performed by the new system and their relationship with each other.
- **Function network** which are similar to function hierarchy but they highlight the those functions which are common to more than one procedure.
- **List of attributes** of the entities - these are the data items which need to be held about each entity (record)

System Design

- Based on the user requirements and the detailed analysis of a new system, the new system must be designed.

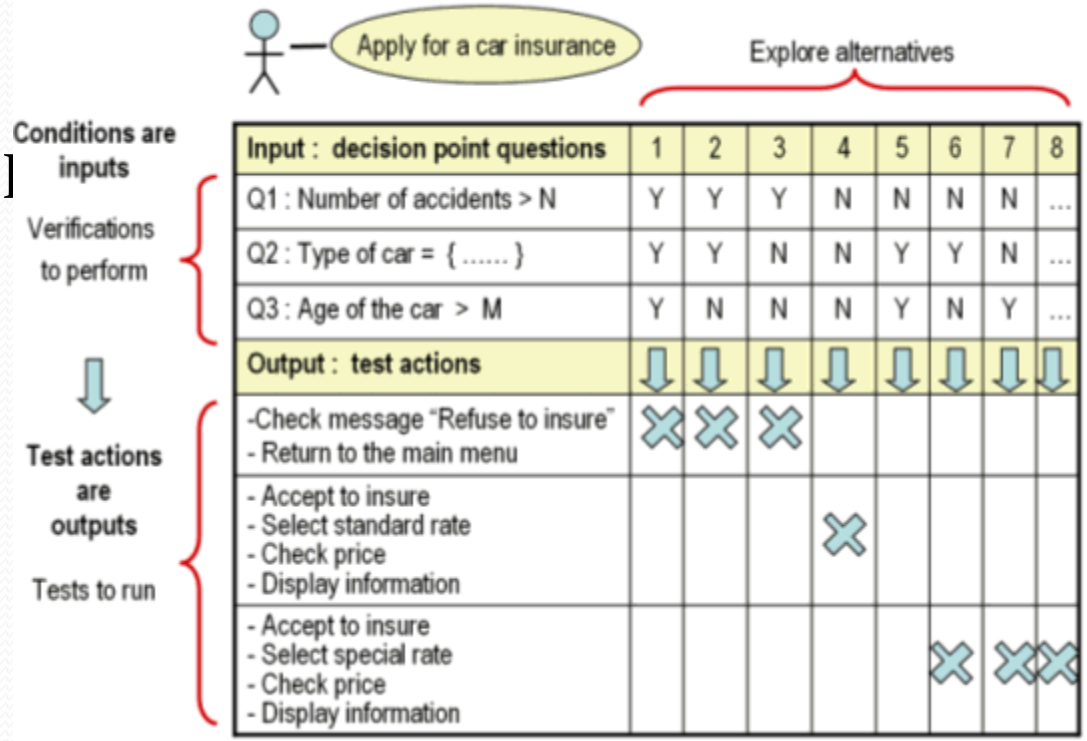
- This is the phase of **system designing**. It is a most crucial phase in the development of a system.

- Normally, the design proceeds in two stages :

  - Preliminary or general design
  - Structure or Detailed design

- There are several tools and techniques used for designing. These tools and techniques are:

- Flowcharts
- Data flow diagrams (DI
- Data dictionary
- Structured English
- Decision trees



Coding

- **Coding** the new system into computer programming language converts human instructions into a format that computer understands.
- Coding is **stage** where defined procedure are transformed into control specifications by the help of a computer language.
- This is also called the **programming phase** in which the programmer converts the program specifications into computer instructions, which we refer as **programs**.
- The programs coordinate the data movements and control the entire process in a system.
- It is generally felt that the programs must be modular in nature. This helps in fast development, maintenance and future change, if required.

Testing

- Removing all the bugs, if any - **Before implementing**
- Test plan is developed and run on given set of test data.
- Output of test run should **match** expected results.
- Using test data following test runs are carried out:
  - **Unit test**: After coding / compiling programs, they are **individually** tested with test data. Any ambiguity must be noted & debugged.
  - **System Test**: done after unit test. Complete system is executed on actual data. Results or output of system is analyzed. If output is not matched with expected outputs, errors are identified and are fixed.

Implementation

- After UAT, the implementation phase begins.
- It is the stage during which theory is turned into practice.
- All programs of the system are loaded onto the user's computer.
- After loading the system, training of the users starts. Main topics of such type of training are:
  - How to execute the package
  - How to enter the data
  - How to process the data (processing details)
  - How to take out the reports
- After the users are trained about the computerized system, manual working has to shift from manual to computerized working.

System Run Strategies:

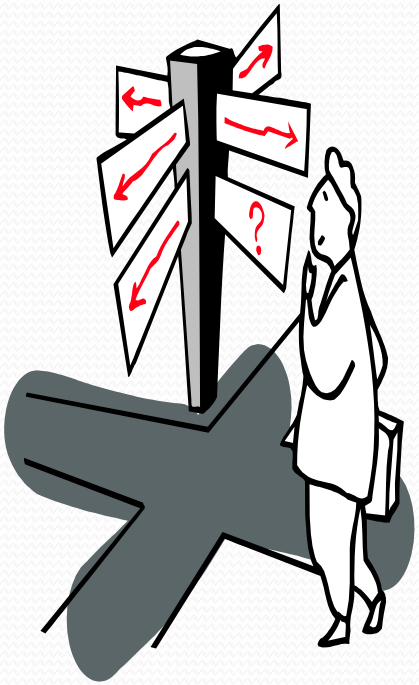
- **Parallel run:** computerized & manual systems are executed in parallel. Advantages of Parallel run:
  - Manual results comparison with the computerized one.
  - Failure of the computerized system at the early stage, does not affect the working of the organization.
- **Pilot run:** New system is installed in parts. Some part of the new system is installed first and executed successfully for considerable time period. Advantages:
  - When results are found satisfactory then only other parts are implemented.
  - This strategy builds the confidence and the errors are traced easily.

Maintenance

- **Maintenance is required to**:
  - eliminate errors in the system during its working life
  - tune the system to any variations in its working environment.
- Errors are always in system that must be corrected.
- **System Review**: is necessary from time to time for:
  - knowing the full capabilities of the system
  - knowing the required changes or the additional requirements
  - studying the performance
- **Major change** during the review:
  - If a major change to a system is needed, a new project may have to be set up to carry out the change.
  - New project will then proceed through all above life cycle phases.

learning Objectives

- Describe and Understand Structured analysis and design
- Describe and Understand **Object-Oriented** analysis and design
- **Distinguish** between structured analysis and design and object-oriented analysis and design



Introduction

SAD (Structured analysis) is a traditional systems development technique that is time-tested and easy to understand.

Structured analysis uses a set of process models to describe a system graphically.

Because it focuses on processes that transform data into information, structured analysis is called a **process-centered technique**

Whereas structured analysis treats **processes and data as separate** components, object-oriented analysis **combines data and the processes** that act on the data into things called objects.

System analysts use O-O to model real world business processes and operation.

The result is a set of software objects that represent actual people, things, transactions and events

Systems Development Methodologies

- Structured analysis and design
  - "Structured analysis is a set of techniques and graphical tools that allow the analyst to develop a new kind of system specification that are easily understandable to the user. Analysts work primarily with their wits, pencil and paper." [Kendall 1996]

Systems Development Methodologies

- Structured analysis and design
  - Developed in the late 1970s by DeMarco, Yourdon, and Constantine after the emergence of structured programming.
  - In 1989, Yourdon published "Modern Structured Analysis".
  - The availability of CASE tools in the 1990s enabled analysts to develop and modify the graphical SAD models.
  - Philosophy of structured analysis and design
    - Analysts attempt to divide large, complex problems into smaller, more easily handled ones. "**Divide and Conquer**"
    - **Top-Down** approach
    - **Functional** view of the problem.
    - Analysts use **graphics to illustrate their ideas** whenever possible
  - Goals of SAD
    - **Improve Quality** and reduce the risk of system failure
    - Establish **concrete requirements** specifications and complete requirements documentation
    - Focus on **Reliability**, **Flexibility**, and **Maintainability** of system

Systems Development Methodologies

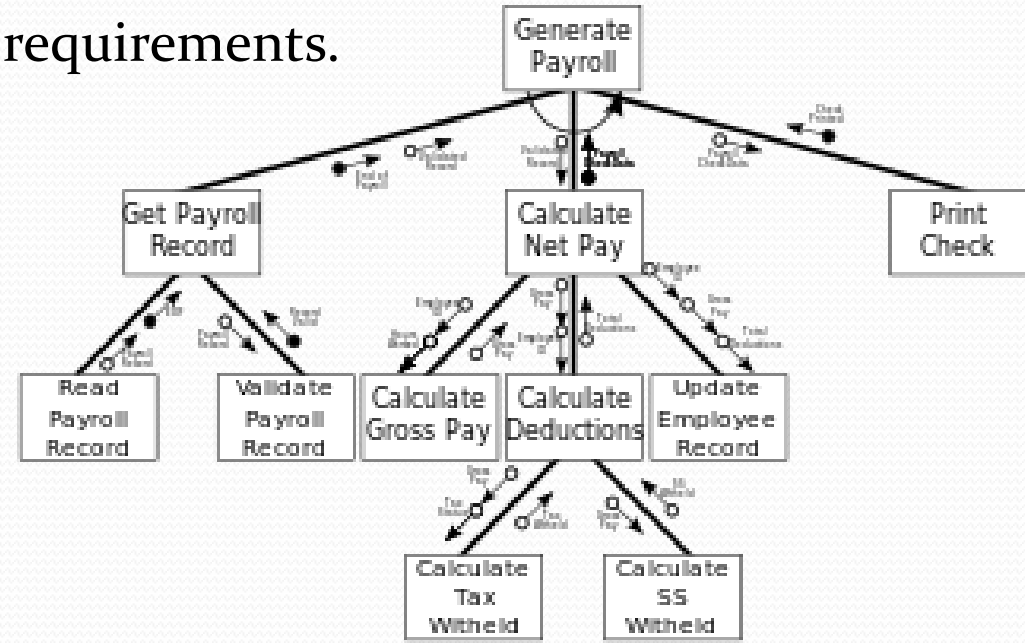
- Structured analysis and design
- Elements of Structured Analysis and Design
  - **Essential Model:** Model of what system must do.
    - Does not define how the system will accomplish its purpose.
    - Is a combination of the **environmental and behavioral** model
      - **Environmental Model**
        - Defines the **scope** of the proposed system.
        - Defines the **boundary** and interaction between the system and the outside world.
        - Composed of: Statement of Purpose, Context Diagram, and Event List.

Systems Development Methodologies

- Structured analysis and design
  - **Behavioral Model**
    - Model of the internal behavior and data entities of the system.
    - Models the functional requirements.
    - Composed of Data Dictionary, Data Flow Diagram, Entity Relationship Diagram, Process Specification, and State Transition Diagram.

Systems Development Methodologies

- Structured analysis and design
  - **Implementation Model**
    - Maps the functional requirements to the hardware and software.
    - Determines which functions should be manual and which should be automated.
    - Defines the Human-Computer Interface.
    - Defines non-functional requirements.
    - Tool: Structure Charts



Systems Development Methodologies

- Structured analysis and design
  - Advantages of Structured analysis and design
    - visual, so it is easier for users/programmers to understand
    - Makes good use of graphical tools
    - A mature technique
    - Process-oriented approach is a natural way of thinking
    - Flexible
    - simple and easy to understand and impalement

Systems Development Methodologies

- Structured analysis and design
  - Disadvantages of Structured analysis and design
    - Not enough user-analyst interaction
    - It depends on dividing system to sub systems but it is to decide when to stop decomposing

Systems Development Methodologies

- Object-Oriented analysis and design
  - Object-Oriented analysis and design becoming popular because of its ability to thoroughly represent complex relationships, as well as represent data and data processing with a consistent notation
  - Object-Oriented analysis and design blend analysis and design in evolutionary process
  - It allows you to deal with the complexity inherent in a real-world problem by focusing on the essential and interesting features of an application

Systems Development Methodologies

- Object-Oriented analysis and design
  - Process of progressively developing representation of a system component (or object) through the phases of analysis, design and implementation
  - The model is abstract in the early stages
  - As the model evolves, it becomes more and more detailed

Systems Development Methodologies

- Object-Oriented analysis and design
  - Object-Oriented systems development life cycle
    - The Object-Oriented development life cycle consists of progressively developing an object representation through three phases analysis, design, and implementation
    - Analysis Phase
    - Object-oriented analysis is a popular approach that sees a system from the viewpoint of the objects themselves as they function and interact
      - Model of the real-world application is developed showing its important properties
      - Model specifies the functional behavior of the system independent of implementation details

Systems Development Methodologies

- Object-Oriented analysis and design
  - Design Phase
    - Analysis model is refined and adapted to the environment
    - Can be separated into two stages
      - System design
        - Concerned with overall system architecture
      - Object design
        - Implementation details are added to system design
  - Implementation Phase
    - Design is implemented using a programming language or database management system

Systems Development Methodologies

- Structured analysis and design vs. object oriented analysis and design

Similarities

- Both SAD and OOAD had started off from programming techniques
- Both techniques use graphical design and graphical tools to analyze and model the requirements.
- Both techniques provide a systematic step-by-step process for developers
- Both techniques focus on documentation of the requirements

Systems Development Methodologies

- Structured analysis and design vs. object oriented analysis and design

Differences

- SAD is Process-Oriented
- OOAD combines data and the processes
- OOAD encapsulates as much of the systems' data and processes into objects, while SAD separates between them