Design Methodologies

- Top-down design focus on the process of transforming input into the output, resulting in a hierarchy of tasks.

- Object oriented design focuses on the data objects that are to be transformed, resulting in a hierarchy of objects.

- "Read the specification of the software you want to build. Underline the verbs if you are after procedural code, nouns if you aim for an object oriented program" -- Grady Booch.

Problem Solving

- Encapsulation – Bundling data and actions so that the logical properties of data and actions are separated from the implementation details.

- Information hiding – The practice of hiding the details of a module with the goal of controlling access to the details of the module.

- Abstraction – A model of a complex system that includes only the details essential to the viewer.

Problem Solving

- Data abstraction – The separation of the logical view of data from its implementation.

- Procedural abstraction – The separation of the logical view of an action from its implementation.

- Control abstraction – The separation of the logical view of a control structure from its implementation.

- Control structure – A statement used to alter the normal sequential flow of control.

Flowcharts

- A flowchart is a type of diagram that represents an algorithm or process, showing the steps as boxes of various kinds, and their order by connecting these with arrows.

- This diagrammatic representation can give a step-by-step solution to a given problem.

- Process operations are represented in these boxes, and arrows connecting them represent flow of control.

- Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

A simple flowchart for computing factorial N (N!)
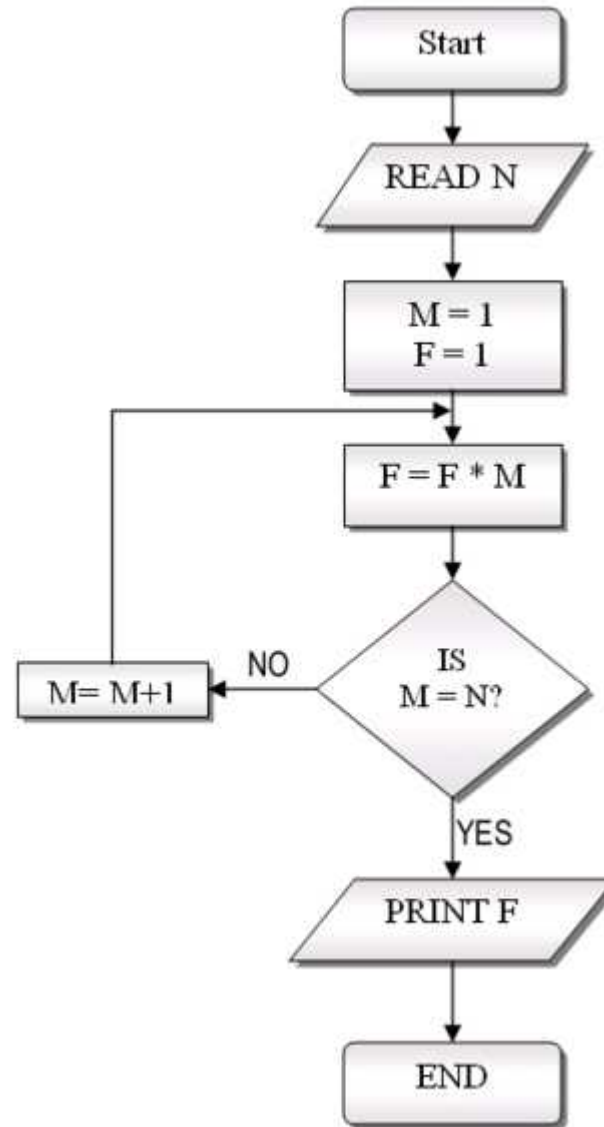
Flowcharts

- Flowcharts can be modeled from the perspective of different user groups (such as managers, system analysts and clerks):

    - Document flowcharts, showing controls over a document-flow through a system

    - Data flowcharts, showing controls over a data-flow in a system

    - System flowcharts showing controls at a physical or resource level

    - Program flowchart, showing the controls in a program within a system

- Modern techniques such as UML activity diagrams can be considered to be extensions of the flowchart.

6

Programming Languages

- Programming language – A set of grammar rules, symbols and special words used to construct a program – that is, to express a sequence of instructions for a computer.

- Program – A sequence of instructions written for a specific task.

- Syntax – The formal grammar rules governing the construction of valid instructions.

- Semantics – The set of rules that gives the meaning of instructions in a language.

Programming Languages

- Programming languages are available with a spectrum of abstraction.

  - Machine Language – the language that comes with the hardware.

  - Assembly Language – the lowest level programming language.

  - High Level Language – Languages with most abstraction.

- Greater abstraction allows more processing to be done from a single statement.

Machine Language

- The language made up of binary coded instructions that is used directly by the computer.

- Instructions are patterns of bits that by physical design correspond to different commands to the machine.

- Each machine language instruction performs only one very low-level task.

- Each small step in a process must be explicitly coded in machine language.

- Very few programs written in machine language today – inefficient use of programmer's time.

Assembly Language

- A low-level programming language in which a mnemonic represents each of the machine language instructions for a particular computer.

- Assembler – A program that translates an assembly language program into machine code.

- Each type of computer has a different machine language resulting in a variety of assembly languages and translators.

Machine and Assembly Language

- The instruction that tells an x86/IA-32 processor to move an immediate 8-bit value into a register.

  - The binary code for this instruction is 10110 followed by a 3-bit identifier for which register to use.

  - The identifier for the AL register is 000, so the following machine code loads the AL register with the data 01100001.

Machine and Assembly Language

- 10110000 01100001

- This binary computer code can be made more human-readable by expressing it in hexadecimal as follows

- B0 61

- The machine code above can be written as follows in assembly language, complete with an explanatory comment if required, after the semicolon

  - MOV AL, 61h   ; Load AL with 97 decimal (61 hex)

Testing

- Test plan – A document that specifies how a program is to be tested.

- Code-coverage (clear-box) testing – Testing a program or sub-program based on covering all statements in the code.

- Data-coverage (black-box) testing – Testing a program or subprogram based on possible input values, treating the code as a black box.

- Test-plan implementation – Using the test cases specified in a test plan to verify that a program outputs the predicted results.