W1-L1

INTRODUCTION TO OOP

TODAY AGENDA

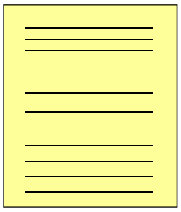
- Programming progression
- Introduction to course and course outline.
  - Course description
  - Course goal
  - Course objective
- Unstructured programming approach
- Problems of unstructured programming approach
- Structured programming approach
- Problems with structured programming approach
- Why we needs OOP
- Characteristic of OOP

PROGRAMMING PROGRESSION…

- Programming has progressed through:
  - machine code
  - assembly language
  - machine-independent programming languages
  - procedures & functions
  - objects

UNSTRUCTURED

- Usually, people start learning programming by writing small and simple programs consisting only of one main program.



- Sequence of instructions which manipulated global data.
- As size increases code becomes more complex to maintain.

DISADVANTAGE OF UNSTRUCTURED APPROACH

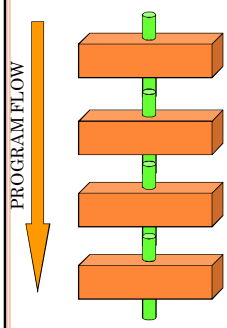
Many disadvantages for large programs

- This programming techniques provide tremendous disadvantages once the program gets sufficiently large.
- Difficult to follow logic
- If something needs to be done more than once must be re-typed
- Hard to incorporate other code
- Not easily modified
- Difficult to test particular portions of the code
- All variable in the program were global,
- Writing , understanding and maintaining long program became a programmer headache.
- …

STRUCTURED PROGRAMMING

- *Structured or procedural programming* attempts to divide the problem into smaller blocks or procedures which interact with other.
- The aim is to clearly define the *structure* of the program before writing program code.

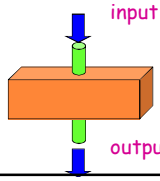
Structured Programming



Ideally each block should be a **black box** –should carry out a well-defined task, interacting with other blocks only through definite inputs and outputs.

Focus on what not on how= functional abstraction , corner stone of SP

Whole emphasis on function not on data



Structured Programming

Strategy

1. Write *pseudo-code* if desired to define the problem.
2. Decide on the program blocks, specifying the inputs and outputs for each blocks.
3. Refine and see if the larger blocks can be in turn sub-divided into smaller blocks.
4. Now you can start programming..just fill in the blocks with computer code !

Structured Programming - Summary

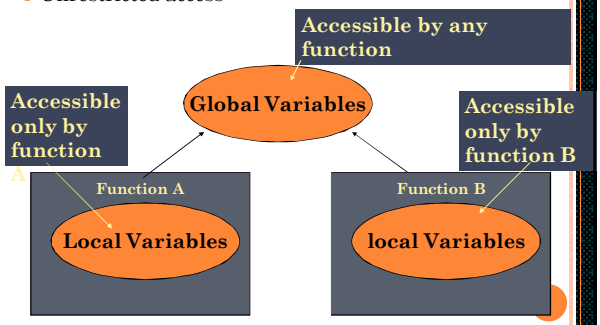
- Structured programs divide the problem into smaller sub-units or blocks, then divided into smaller blocks.. eventually reaching the level of program code.
- The blocks should ideally be self-contained – interactions with other blocks should be explicit
- ALL programming languages support Structured Programming to some extent
- Useful model for small-medium sized projects.
- Becomes unmanageable for larger, more complex projects, with many programmers involved → Object Oriented Programming (OOP).

Limitation of Procedural Language

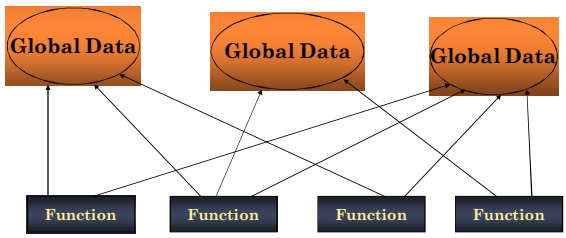
- Procedural paradigm provides very poor modeling of real world
- In order to access the same data by more than one function the data must be made global.
- But in large programs there are many functions and many global data items.

Problems with Structured Approach

- Unrestricted access



Problems with Structured Approach



Limitation of Procedural Language

- The most important problem with procedural paradigm is that its arrangement of separate data and function does a poor job of modeling things in the real world.
- In the physical world we deal with objects such as people and cars.
- Such objects are not like data and they are not like functions.
- Complex real world objects have both attribute and behavior.

Limitation of Procedural Language

- Attribute in the real world are equivalent to data in programming.
- Attribute for people are eye color, height, width etc. and have certain specific values like blue 50, 36 inches.
- Behavior is like a function, you call a function to do some thing while the behavior of real world objects does in response to some stimulus.

Limitation of Procedural Language

- Traditional language give no facility to create your own data type.
- Being able to create your own data type is called extensibility.
- You can extend the capabilities of the language but traditional / procedural language are not usually extensible.

Why do we need OOP

- OOP was developed because limitation were discovered in earlier approaches to programming.