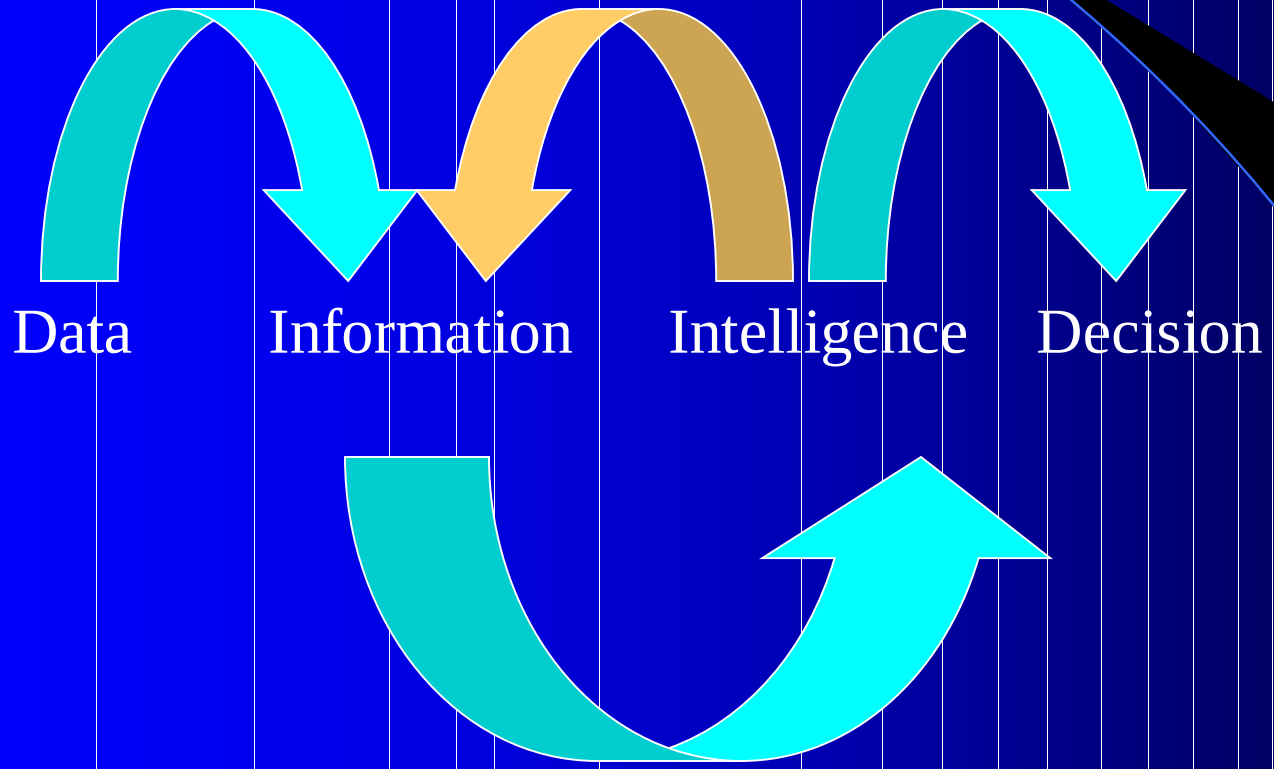
773.707 Software Design Concepts

Lecture III

Modeling the Problem

- Why write computer programs to solve problems?
  - Reduce repetitive work
  - Increase efficiency and speed
  - Reduce error
  - Standardization
  - Minimize training
  - Lower cost

Problem Solving and Decision Making cont'd.



Problem Solving in Everyday Life

- Problem Solving steps:
- Identify the problem - what are we solving
- Understand the problem – constraints, facts, and desired outputs
- Identify solutions
- Select appropriate solutions – identify the steps
- Evaluate the correctness

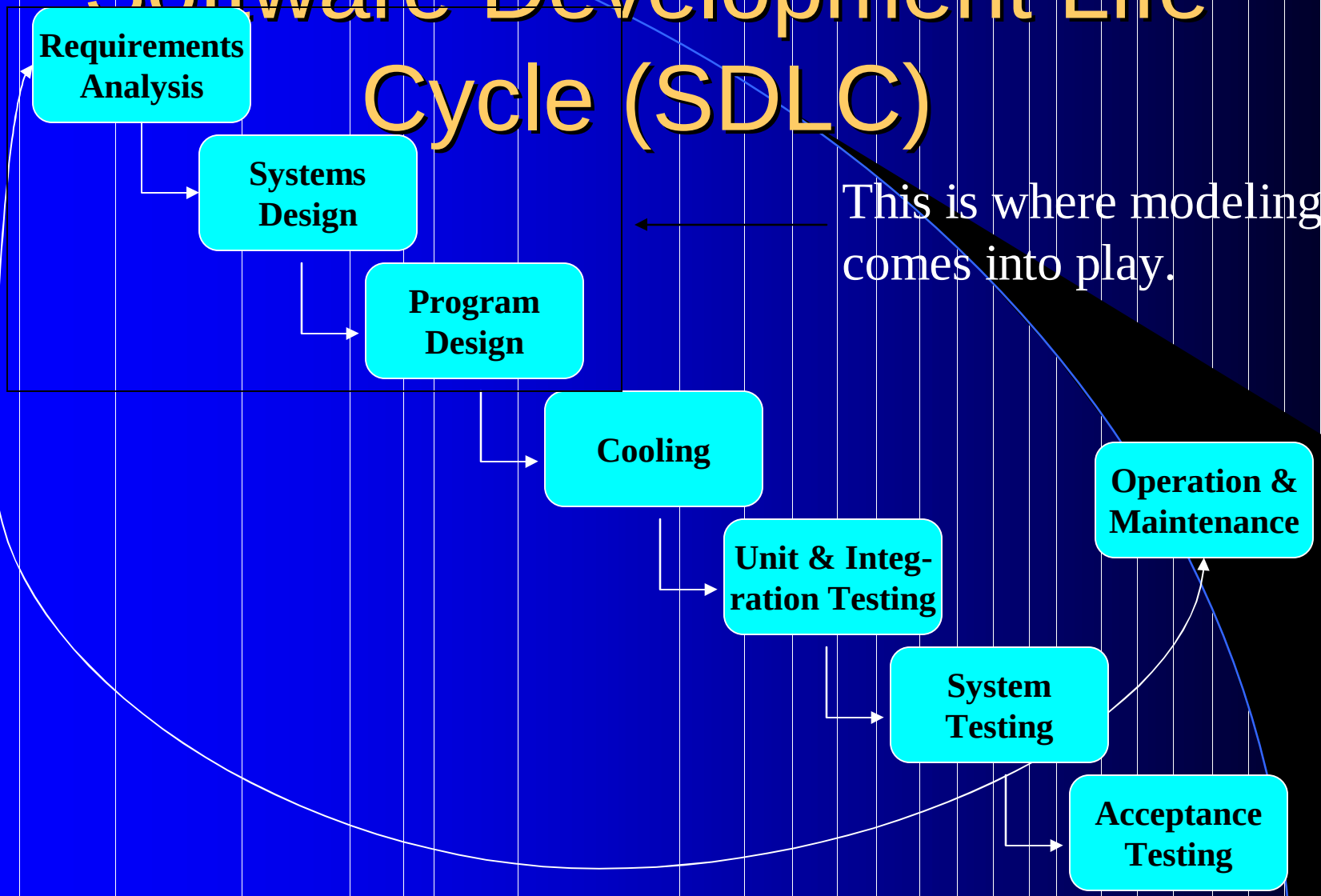
Problem Solving and Decision Making

- Identify the problem – describe what you have, what you want, and what needs be one
- Model the problem – describe the problem (abstraction) and solve the problem
- Build the model

Problem Solving and Decision Making contd.

- Use appropriate tools to model – modeling language, programming language and data structures
- Validate and test

Software Development Life Cycle (SDLC)



This is where modeling comes into play.

Modeling Problem

- Modeling identifies the key components of the problem – the relationships and dependencies
- Developing a model prior to its construction is essential – like a blueprint for a buliding

Unified Modeling Language (UML)

- UML is a language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. UML

Unified Modeling Language Contd.

- UML is a language that specifies modeling elements, notation, and usage guidelines.
- UML is based on object-oriented paradigm
- UML is notational rich, it has extensive use of diagrams.
- UML has classes, objects, sequence, state charts, activity, component, and deployment diagrams.

Unified Modeling Language Contd.

- UML is a complicated modeling language for system engineering.
- To learn more about UML refer:
- The Object Management Group: http://www.omg.org
- Relational Software Corporation: http://www.rational.com/uml
- UML in a nutshell SinansiAlhir, O' Reilly 1998

Programming Concepts (Chapter 3)

- Data Storage, Programs, Charts
- Problem Analysis Chart (PAC)
- Structure Charts (SC)
- Input-Processing-Output (IPO, HIPO)
- Algorithm, Flowchart, Testing a solution

Data Storage

- Run time data is stored in internal memory (short-tem storage)
- Externally data is stored in files: long term
- Program files – set of programming instructions
- Data Files: Data structure or data streams

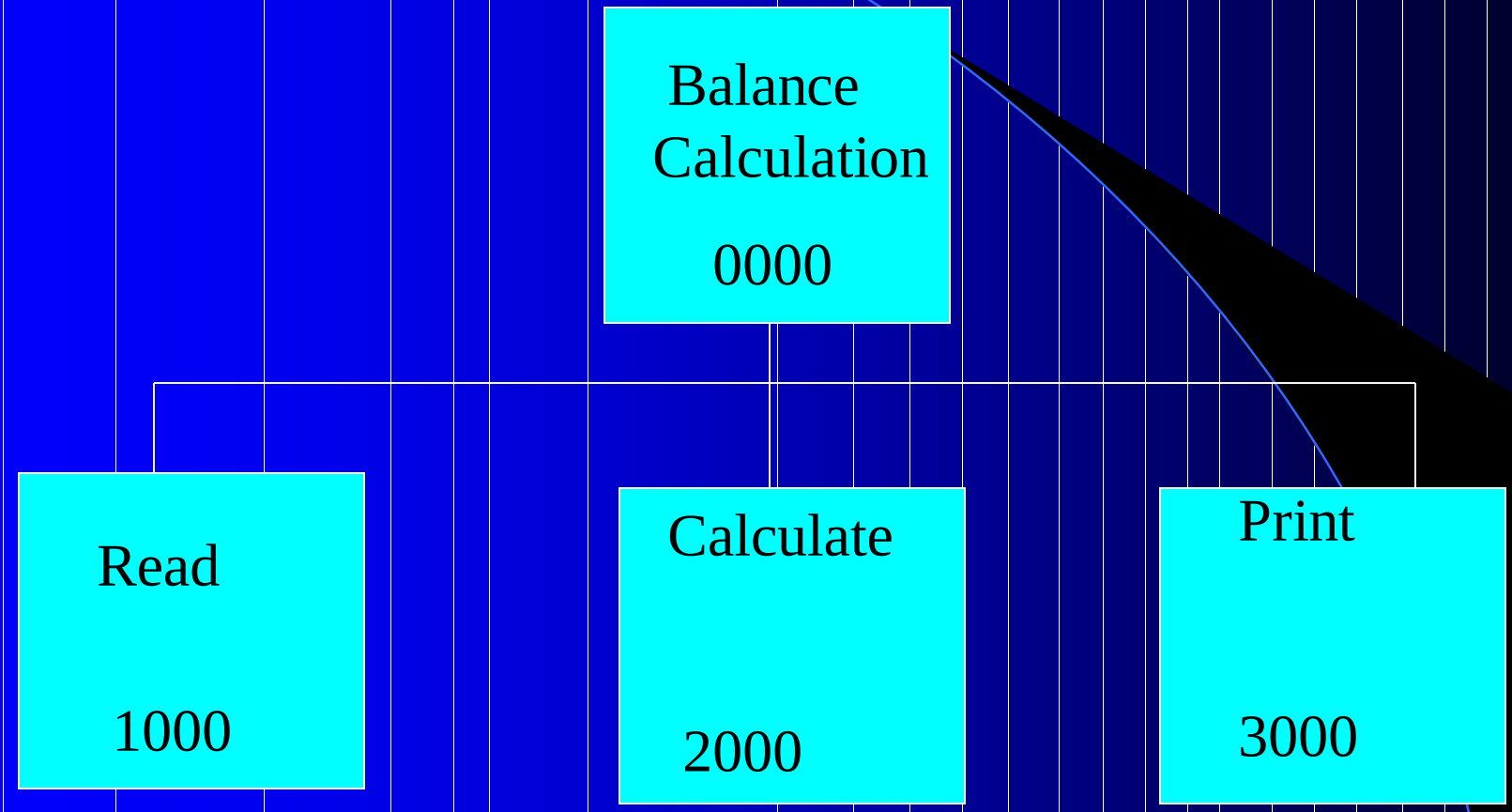
Programs

- Data – fields, Record, File, Database
- Program – a sequence of programming statements
- Algorithm – A sequence of statements for a particular task.

Charts

- Flowcharts
- IPO Charts
- HIPO Charts
- PERT Charts
- GANTT Charts

Examples

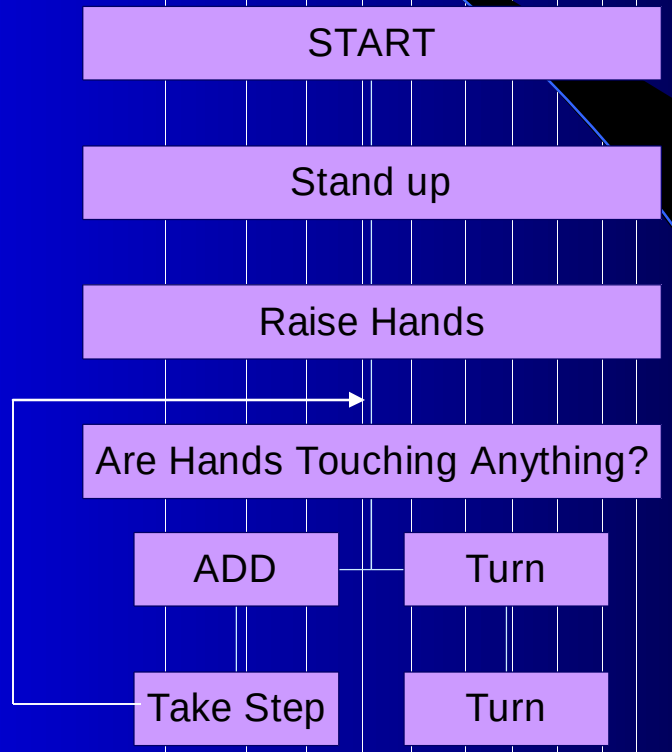


Example

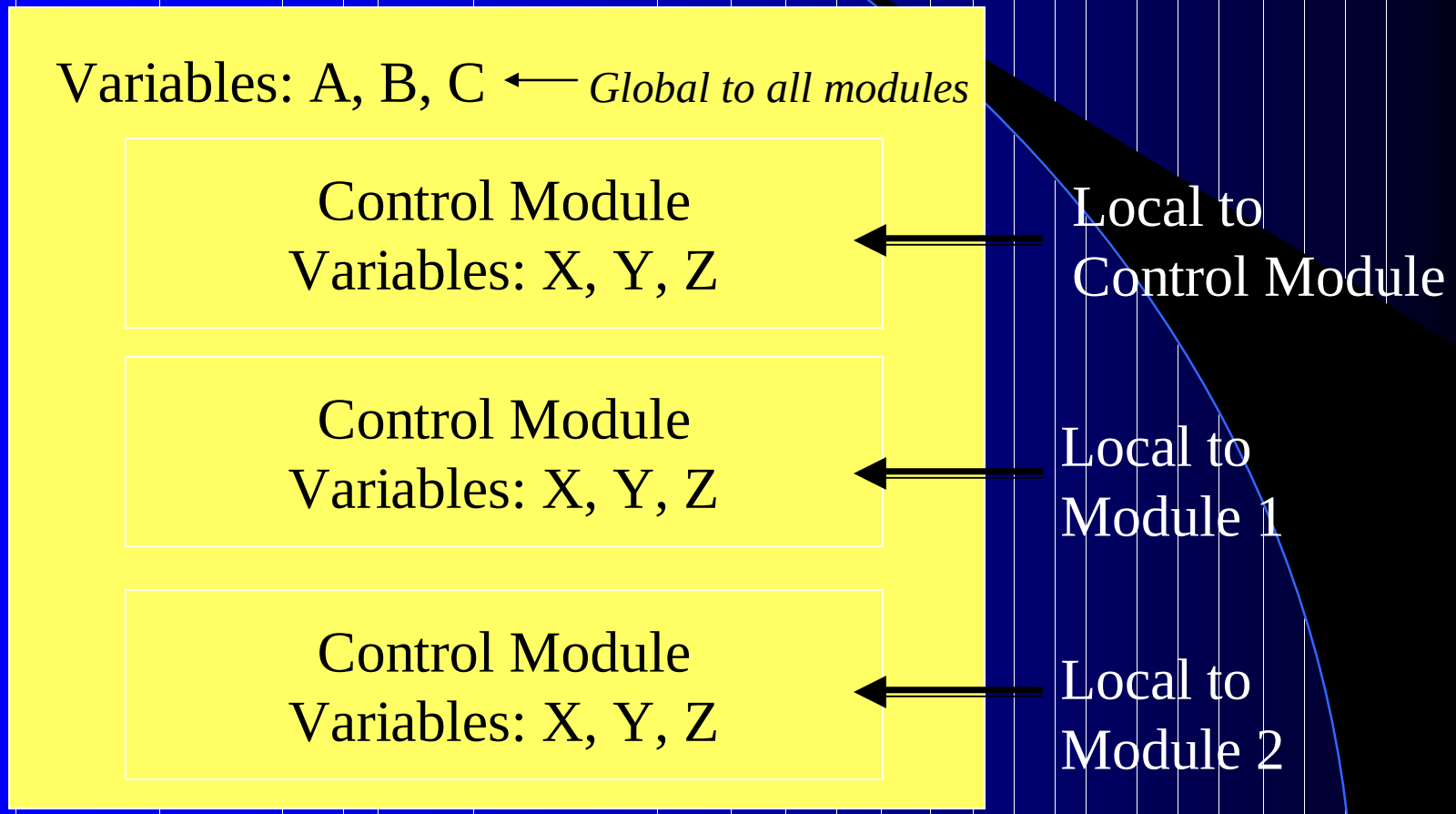
Input Processing Module Reference Number Output

Example

Algorithm:
Start
1. Stand up.
2. Raise Hands
3. Are hands touching anything?
4. Add.
5. Take step.
6. GO TO 3.
7. Turn
8. Turn
...



Example



Example

A = 3
B = 4
B = 5
…

Module1 (A,B,C)

Actual Parameter

Module1 (X,Y,Z)

Formal Parameter

X = Y + Z
Print X