Object Oriented Programming (IT- 2005)

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

School of Computer Engineering

Mr. Abhaya Kumar Sahoo
Assistant Professor [II]
School of Computer Engineering,
Kalinga Institute of Industrial Technology (KIIT),
Deemed to be University, Odisha

3 Credit

Lecture Note 01

Lecture Contents

- ❑ Evolution of programming paradigm
- ❑ Monolithic Programming
- ❑ Procedure oriented programming
- ❑ Object-oriented programming (OOP)
- ❑ C vs. C++

Introduction

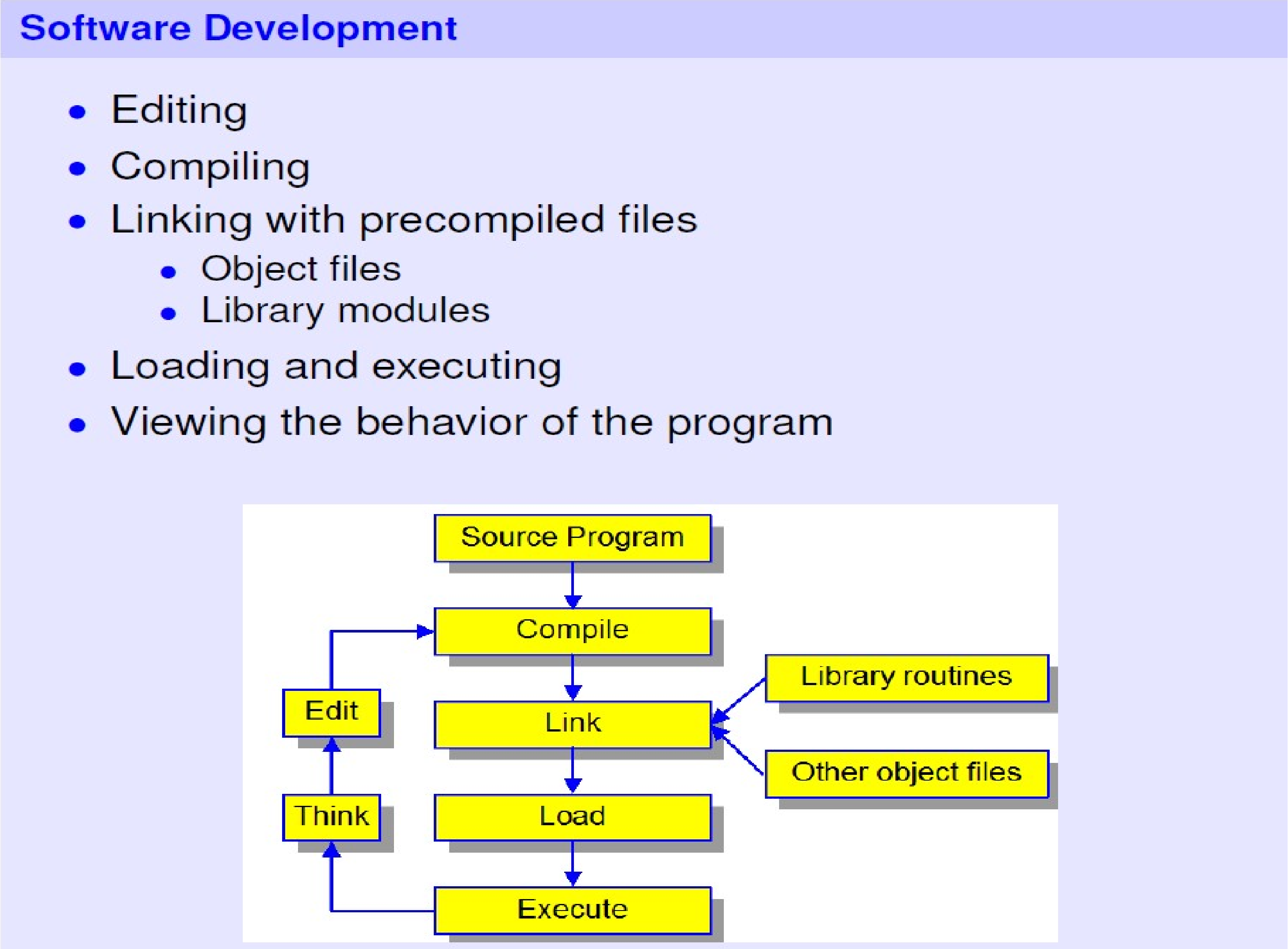
- A programming language is a language specifically designed to express computations that can be performed by the computer.
- Programming languages are used to create programs that control the behavior of a system, to express algorithms, or can be used as a mode of human communication.
- The term ‘programming language’ usually refers to high-level languages such as **BASIC, C, C++, COBOL, FORTRAN, ADA, and PASCAL,** to name a few.
- While high-level programming languages are easy for us to read and understand, the computer understands the machine language that consists only of numbers

Continued…

The determination of the language is dependent on the **following factors:**

- ❑ The type of computer (microcontroller, microprocessor, etc.) on which the program has to be executed.
- ❑ The type of program (system program, application program, etc.).
- ❑ The expertise of the programmer. That is, the proficiency level of a programmer in a particular language.

Software Development



Software Development

- Editing
- Compiling
- Linking with precompiled files
  - Object files
  - Library modules
- Loading and executing
- Viewing the behavior of the program

Programming Paradigms

❑ A programming paradigm is a fundamental style of programming that defines how the structure and basic elements of a computer program will be built. The style of writing programs and the set of capabilities and limitations that a particular programming language has depends on the programming paradigm it supports.

❑These paradigms, in sequence of their application, can be classified as follows:-

- Monolithic programming—emphasizes on finding a solution
- Procedural programming—lays stress on algorithms
- Structured programming—focuses on modules
- Object-oriented programming—emphasizes on classes and objects
- Logic-oriented programming—focuses on goals usually expressed in predicate calculus
- Rule-oriented programming—makes use of 'if-then-else' rules for computation
- Constraint-oriented programming—utilizes invariant relationships to solve a problem

Monolithic Programming

- Here, the program is written with a single function. A program is not divided into parts, hence named as Monolithic Programming.

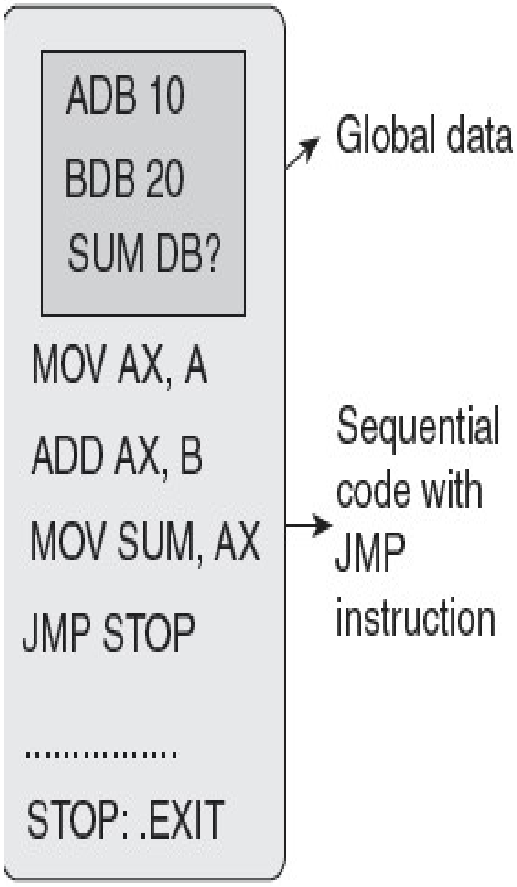
- When the program size increases, it leads to difficulty.

- Here, the data variables declared are global and the statements are written in sequence.

- The program contains jump statements such as goto.

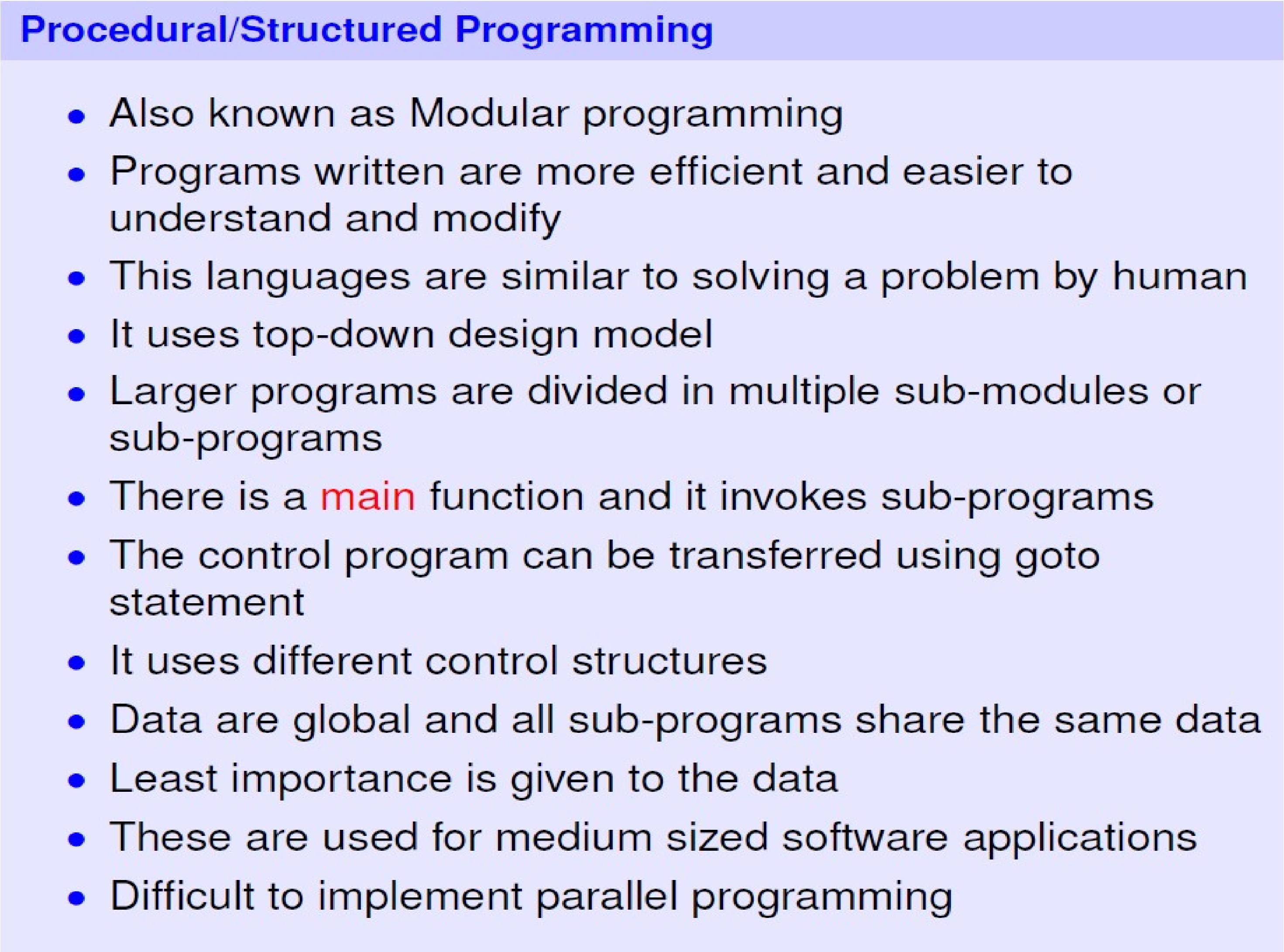
•The concept of subprograms doesn’t exist, hence useful for small programs.

.



```
ADB 10
BDB 20
SUM DB?
MOV AX, A
ADD AX, B
MOV SUM, AX
JMP STOP
...............
STOP: .EXIT
```

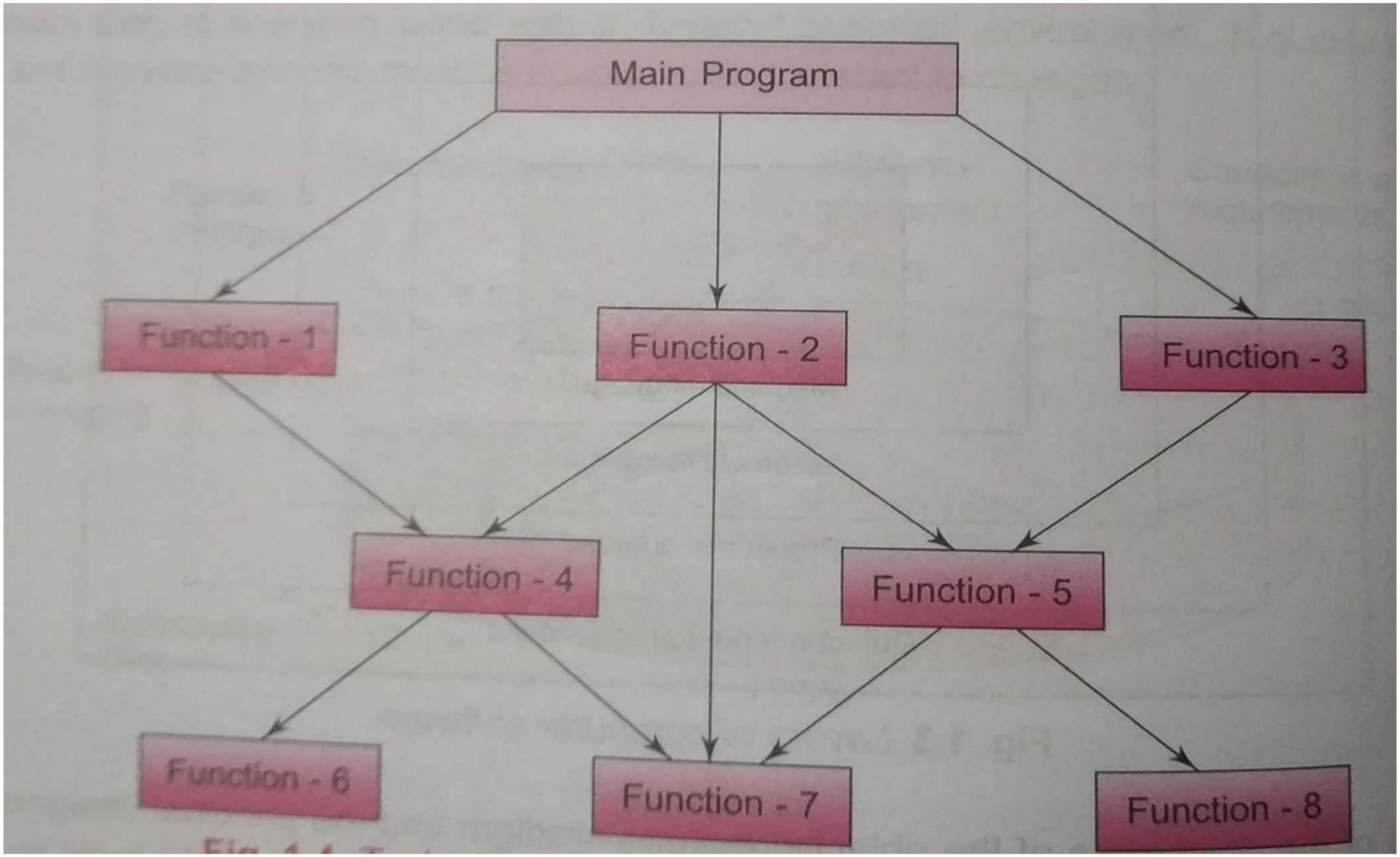
Procedure Oriented Programming



Procedural/Structured Programming

- Also known as Modular programming
- Programs written are more efficient and easier to understand and modify
- This languages are similar to solving a problem by human
- It uses top-down design model
- Larger programs are divided in multiple sub-modules or sub-programs
- There is a main function and it invokes sub-programs
- The control program can be transferred using goto statement
- It uses different control structures
- Data are global and all sub-programs share the same data
- Least importance is given to the data
- These are used for medium sized software applications
- Difficult to implement parallel programming

Typical Structure of POP



Characteristics of POP

- ✓ POP follows a top-down approach.
- ✓ Functions are more important than data in a program.
- ✓ Emphasis is on doing things(algorithms)
- ✓ Large programs are divided in to smaller programs known as functions.
- ✓ Most of the functions shares global data.
- ✓ Data moves openly around the system from function to function.
- ✓ Functions transfer data from one form to another.

Relationship of Data and Function in POP

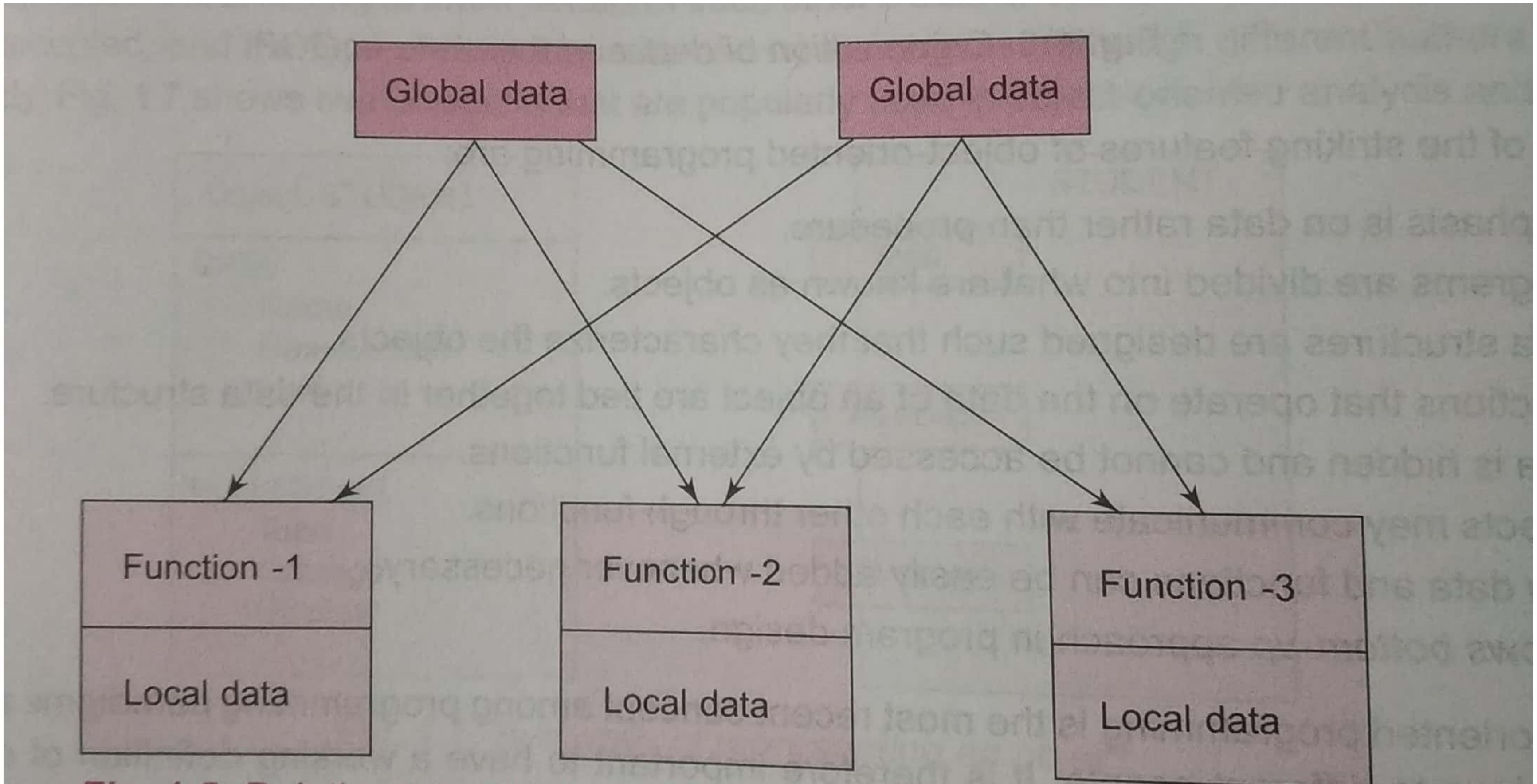


Fig. 1.5 *Relationship of data and functions in procedural programming*

Advantages and Drawbacks of POP

- ✓ **Advantages**
- ✓ The only goal is to write correct programs.
- ✓ Programs were easier to write as compared to monolithic programming.

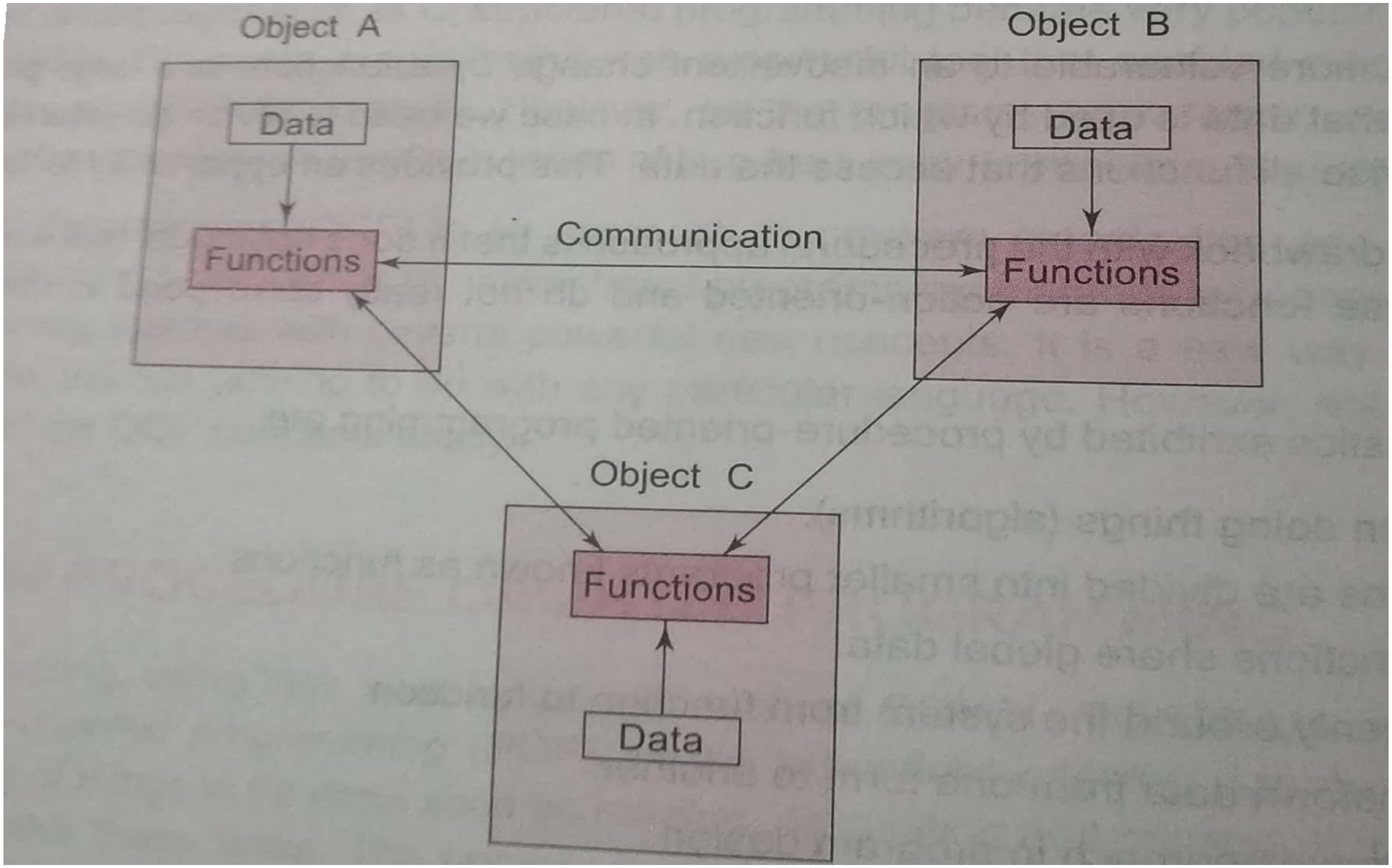
Drawbacks of POP

- ✓ Global data are more vulnerable to an inadvertent change by a function.
- ✓ Writing programs is complex.
- ✓ No concept of reusability.
- ✓ Requires more time and effort to write programs.
- ✓ Programs are difficult to maintain.
- ✓ Global data is shared and therefore may get altered (mistakenly).

Object-Oriented Programming

- ❑ Object-oriented programming (OOP) refers to a type of computer programming (software design) in which programmers define not only the data type of a data structure, but also the types of operations (functions) that can be applied to the data structure.
- The major motovating factor in the invention of Object oriented approach is to remove some of flaws encountered in POP.
- OOP treats data as a critical element in the program development and does not allow it to flow freely around the system.
- It ties data more closely to the functions that operate on it, and protects it from accidential modification from out side functions.
- OOP allows decomposition of a problem into number of entites called objects and then builds data and functions around these objects.

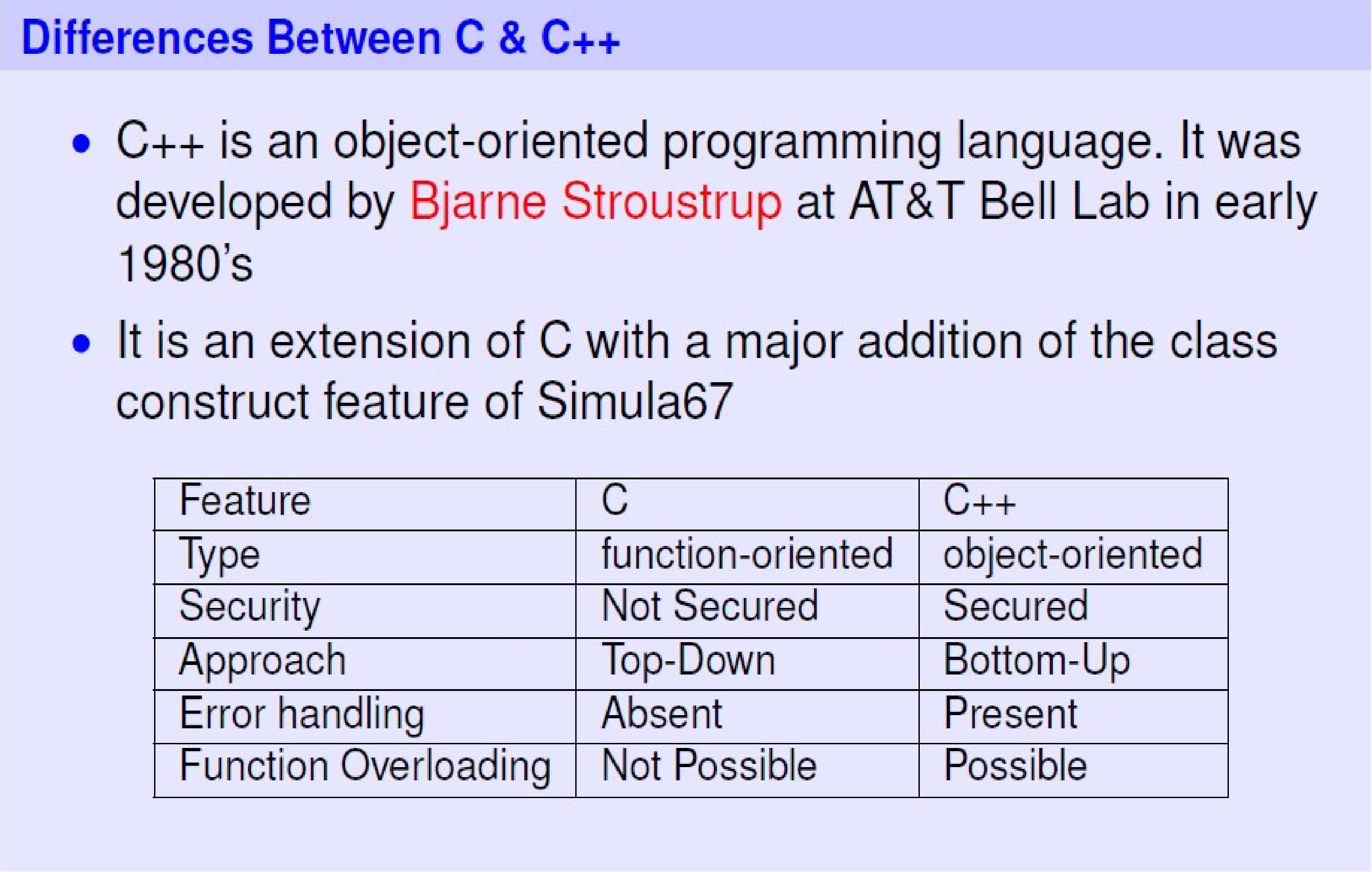
Organization of Data and Functions in OOP



Characteristics of OOP

- Emphasis is on data rather than procedure.
- Programs are divided into what are known as objects.
- Data structures are designed such that they characterize the objects.
- Functions that operate on the data of an object are tied together in the data structure.
- Data is hidden and cannot be accessed by external functions.
- Object may communicate with each other through functions.
- New data and functions can be easily added whenever necessary.
- Follows Bottom-up approach in program design.

C vs. C++



Differences Between C & C++

- C++ is an object-oriented programming language. It was developed by Bjarne Stroustrup at AT&T Bell Lab in early 1980's
- It is an extension of C with a major addition of the class construct feature of Simula67

Feature	C	C++
Type	function-oriented	object-oriented
Security	Not Secured	Secured
Approach	Top-Down	Bottom-Up
Error handling	Absent	Present
Function Overloading	Not Possible	Possible

THANK YOU!