1st Semester
MTCE 601A
COMPUTER SYSTEM SOFTWARE

LECTURE-1

Syllabus Introduction

- 1.1 Introduction to Object Oriented
- 1.2 Introduction to UML
- 1.3 Software Process and OOA&D
- 1.4 Component and CBSD
- 1.5 Patterns and Architecture

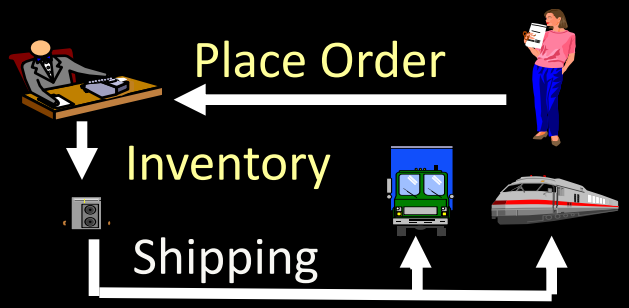
1.1 Introduction to Object-Oriented

- OO Programming (procedural V.S. OO)
- Basic concepts of OO

OO Programming

Designing Programs

Software Development – Solving Problem



Problem Space

Descriptions of problem (Human: Requirements)

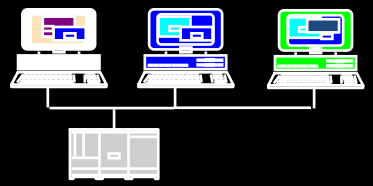
Business Process

Natural Language

A Gap between languages

Descriptions of solution (Human: Designing Program)

Programming Language

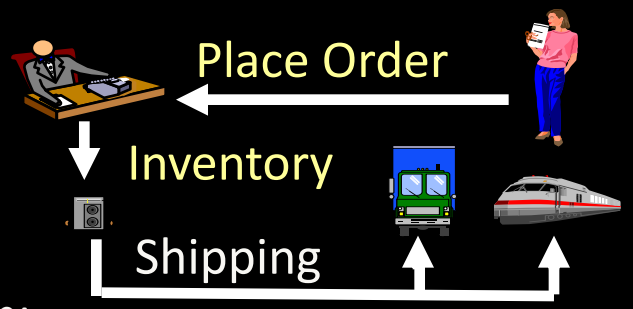


Execution of program

Computer System

Solution Space

Software Development – Solving Problem



Problem Space

Descriptions of problem (Human: Requirements)

Business Process

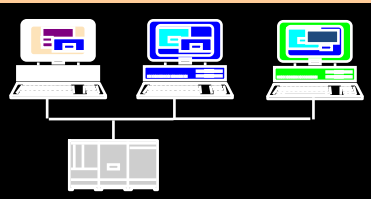
Natural Language

A Gap between languages

Programming Language

Descriptions of solution (Human: Designing Programs)

- **High-Level Language** (Object-Oriented) e.g. C++ Java
- **High-Level Language** (Procedural) e.g. C, BASIC
- **Assembly Language**
- **Machine Language**



Execution of program

Computer System

Solution Space

Procedural Programming

- This programming paradigm is essentially an abstraction of machine /assembly language.
- Program is organized around procedures.
- Focus on data structures, algorithms and sequencing of steps

Programs = Algorithm + Data Structure

An algorithm is a set of instructions for solving a problem

A data structure is a construct used to organize data in a specific way.

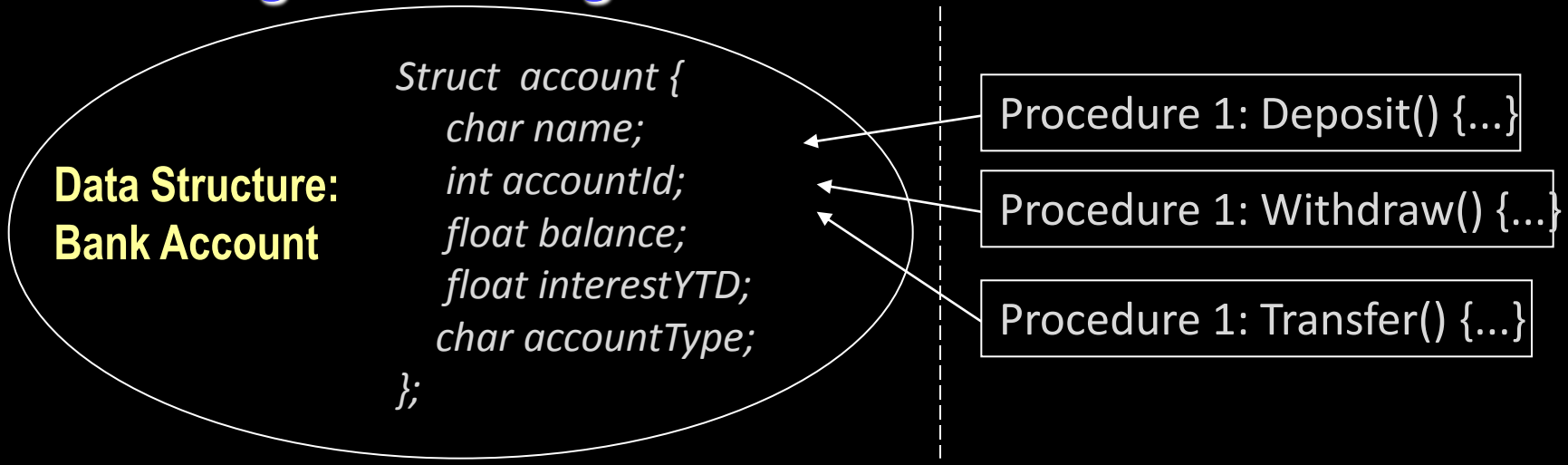
Most computer languages, from early examples like FORTRAN and ALGOL to more recent languages like C and Ada, have been imperative or procedural.

Procedural Programming - Example

- **Writing a program to handle bank accounts**
  - Customer can open different type of accounts, such as cash account, check account and Loan account.
  - For each account, customer can deposit, withdraw or transfer.
- **How to write this program with C ?**

Procedural Programming - Example

Programs = Algorithm + Data Structure



- **A procedural programming language usually consists of :**
  - A collection of , each of which at any stage contains a certain value (a number, a character, a string of characters, etc)
  - A collection of that change the values of these variables.
- **The building-block of this type program is the *procedure* or *function*.**

Procedural Programming - Disadvantages

- **Procedures and data are clearly separated.**
- **Transformation of concepts between analysis & implementation.**
- **Design models are a long step from implementation.**
- **Procedures are often hard to reuse.**
- **Programs are often hard to extend and maintain.**

Object-Oriented Programming: OOP

- A design and programming technique
- Some terminology:
  - *object* - usually a person, place or thing (a noun)
  - *method* - an action performed by an object (a verb)
  - *type* or *class* - a category of similar objects (such as *automobiles)*
- Objects have both data and methods
- Objects of the same class have the same data elements and methods
- Objects send and receive ***message****s* to invoke actions

Object-Oriented Programming - Example

- **Writing a program to handle bank accounts**
  - Customer can open different type of accounts, such as cash account, check account and Loan account.
  - For each account, customer can deposit, withdraw or transfer.
- **How to write this program with C++ or Java ?**

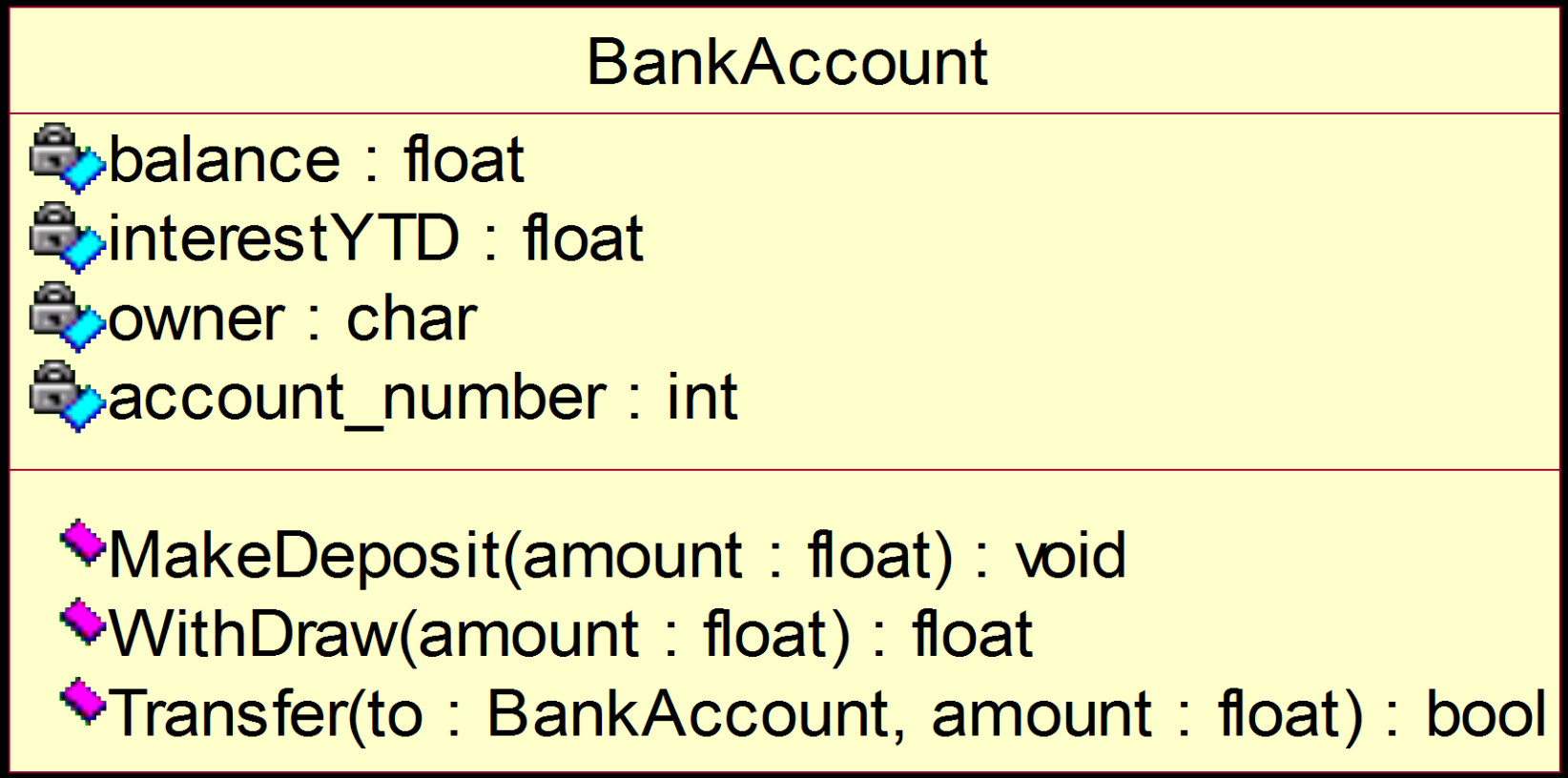
Object-Oriented Programming - Example

- **Object-Oriented approach**
  - combine the accounts (data) with the operations on the accounts to objects.
  - A new kind of data type: BankAccount class
- **C++ code:**

```
Class BankAccount {
  private:
    float balance;
    float interestYTD;char * owner;
    int account_number;
  public:
    void Deposit (float amount) {...}
    float WithDraw (float amount) {…}
    bool Transfer (BankAccount  & to,  float amount) {…}
};
```

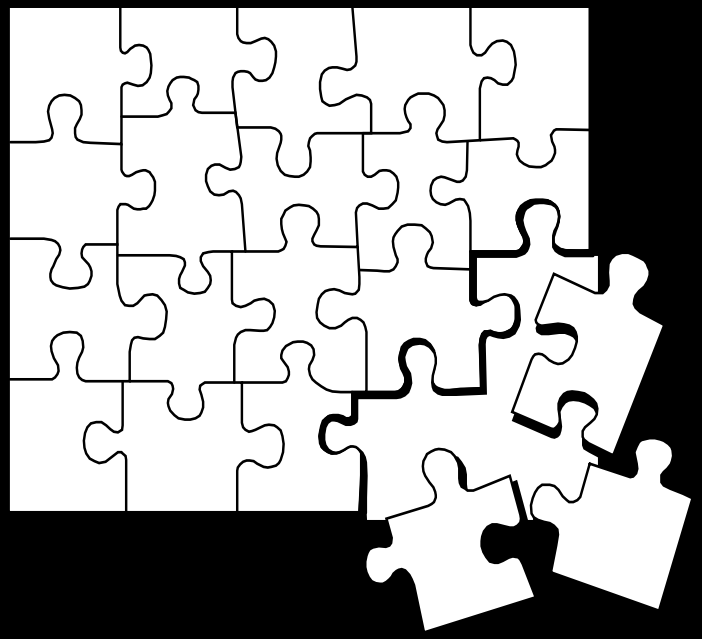
Object-Oriented Programming - Example

- **The building-block of this type program is class or objects.**



What Is Object Technology?

- Object Technology
  - A set of principles guiding software construction together with languages, databases, and other tools that support those principles. (*Object Technology - A Manager's Guide*, Taylor, 1997)



The History of Object Technology

- Major object technology milestones