

C++ Programming Essentials

Unit 1: Sequential Programming

CHAPTER 1: INTRODUCTION

DR. ERIC CHOU

IEEE SENIOR MEMBER

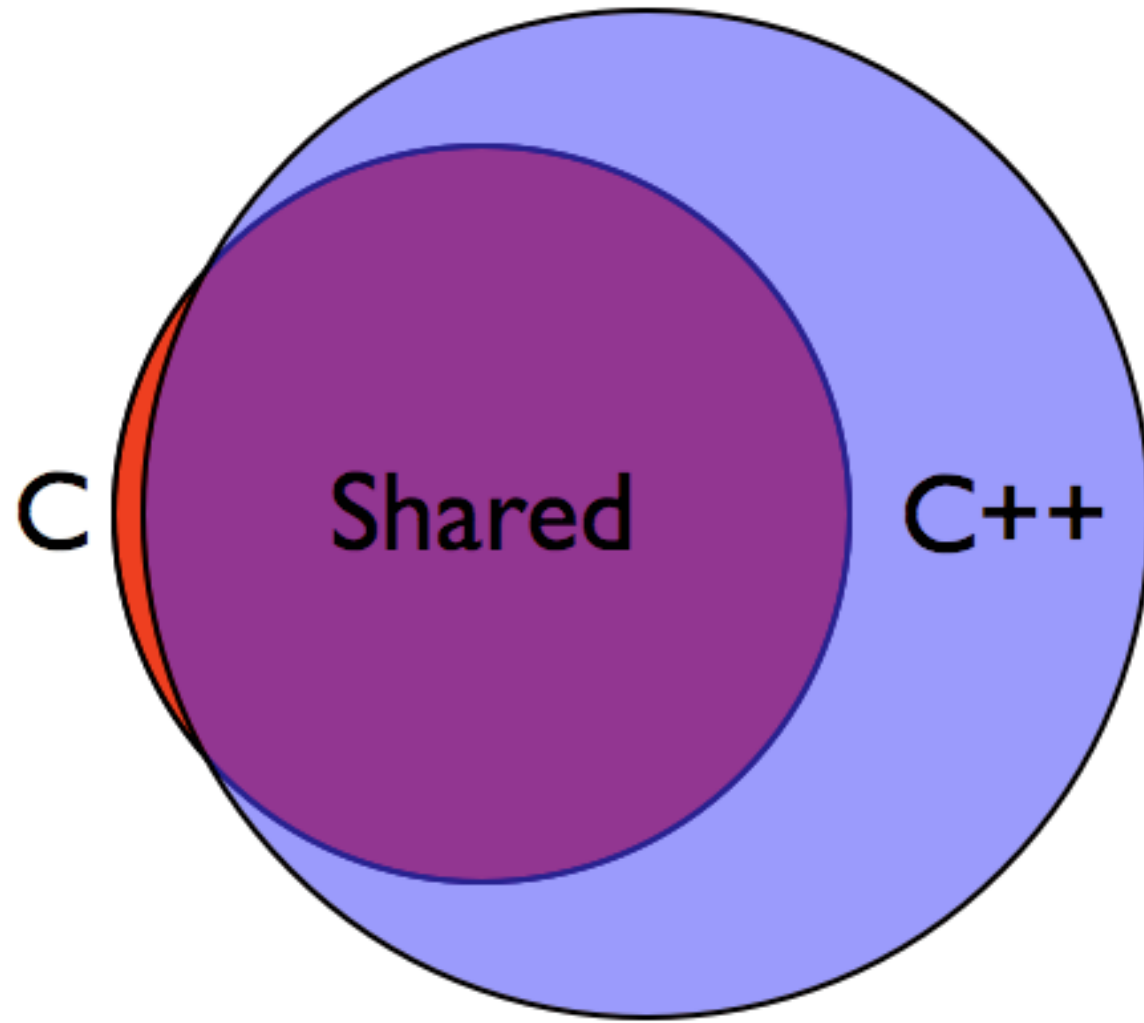
LECTURE 1

Overview of This Course



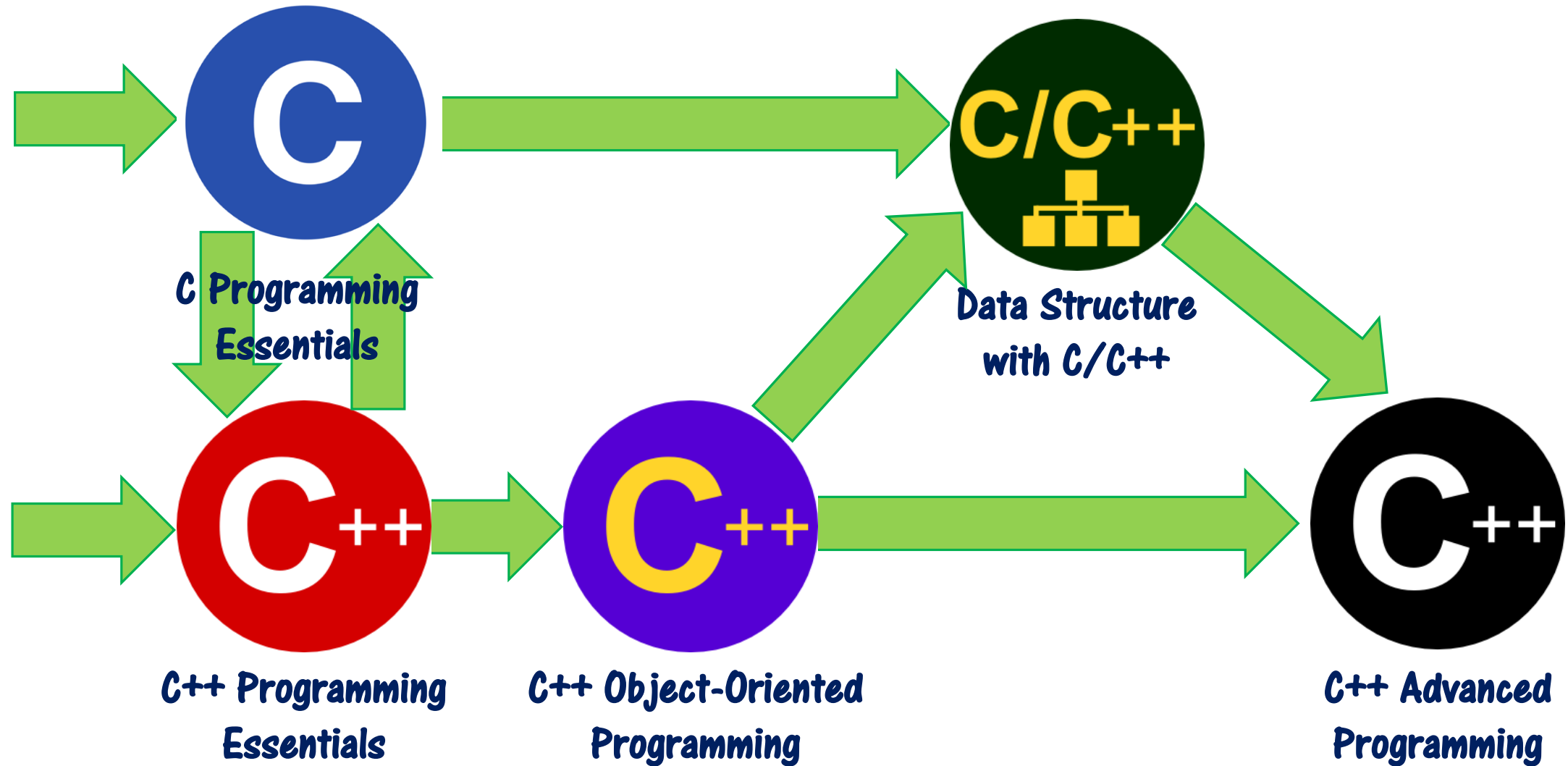
C++ Programming Language

- C++ is a general-purpose object-oriented programming (**OOP**) language, developed by **Bjarne Stroustrup**, and is an extension of the C language.
- It is therefore possible to code **C++** in a "C style" or "object-oriented style." In certain scenarios, it can be coded in either way and is thus an effective example of a hybrid language.
- **C++** is considered to be an intermediate-level language, as it encapsulates both high- and low-level language features.
- Initially, the language was called "**C with classes**" as it had all the properties of the C language with an additional concept of "classes." However, it was renamed C++ in 1983.
- It is pronounced "see-plus-plus."



The
Relationship
between C and
C++

C/C++ Course Series



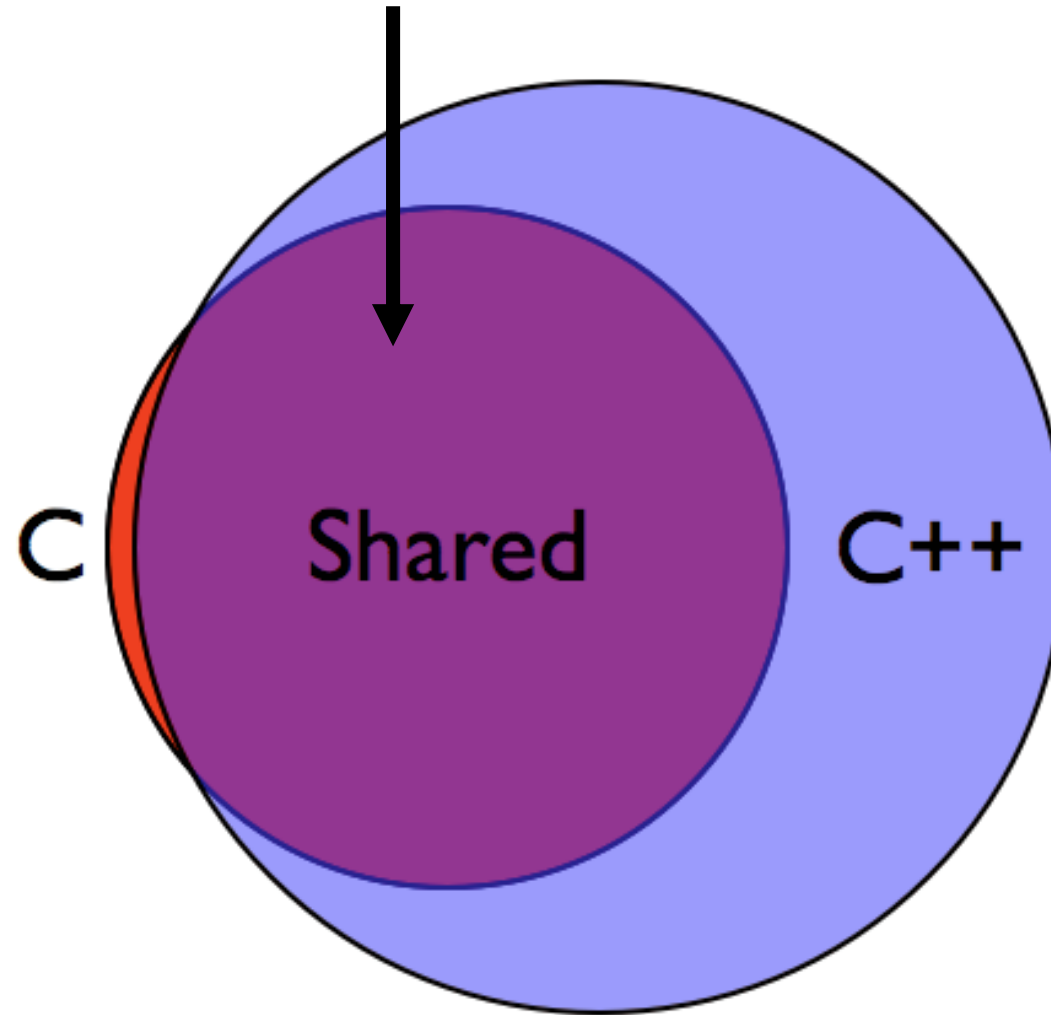
C++ Programming Essentials Course

Unit 1: Sequential Programming
(Elementary Programming)

Unit 2: Structured Programming

Unit 3: Basic Abstract Data Types

Programming Essentials



Target Audience

- College or high school students who want to learn **C++** programming. This is the beginner's class.
- Working individuals who want to sharpen their **C/C++** programming skills.

Goals

- Student will be able to write basic programs in **C++** programming language using **Dev C++** IDE and with gcc compiler.
- Match the college level **C++ 1** course programming skill level.

LECTURE 2

Introduction to Computer Science: Hardware and Software

What is Programming?

*Planning or scheduling
the performance of a
task.*

Consciously thinking
about each step

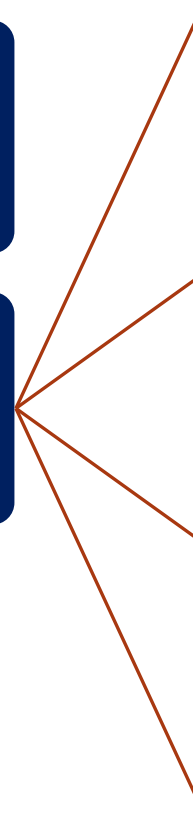
Example: Accelerating
in a car

Move right foot to gas
pedal

Apply pressure to gas
pedal with right foot

If speed is too high,
apply less pressure.

If speed is too low,
apply more pressure.





Are Computers Intelligent?

Do we really need to be involved in programming computers?

- They have beaten world chess champions.
- They help predict weather patterns.
- They can perform arithmetic quickly.

So, a computer has an IQ of _____.



What Do We Have To Do?

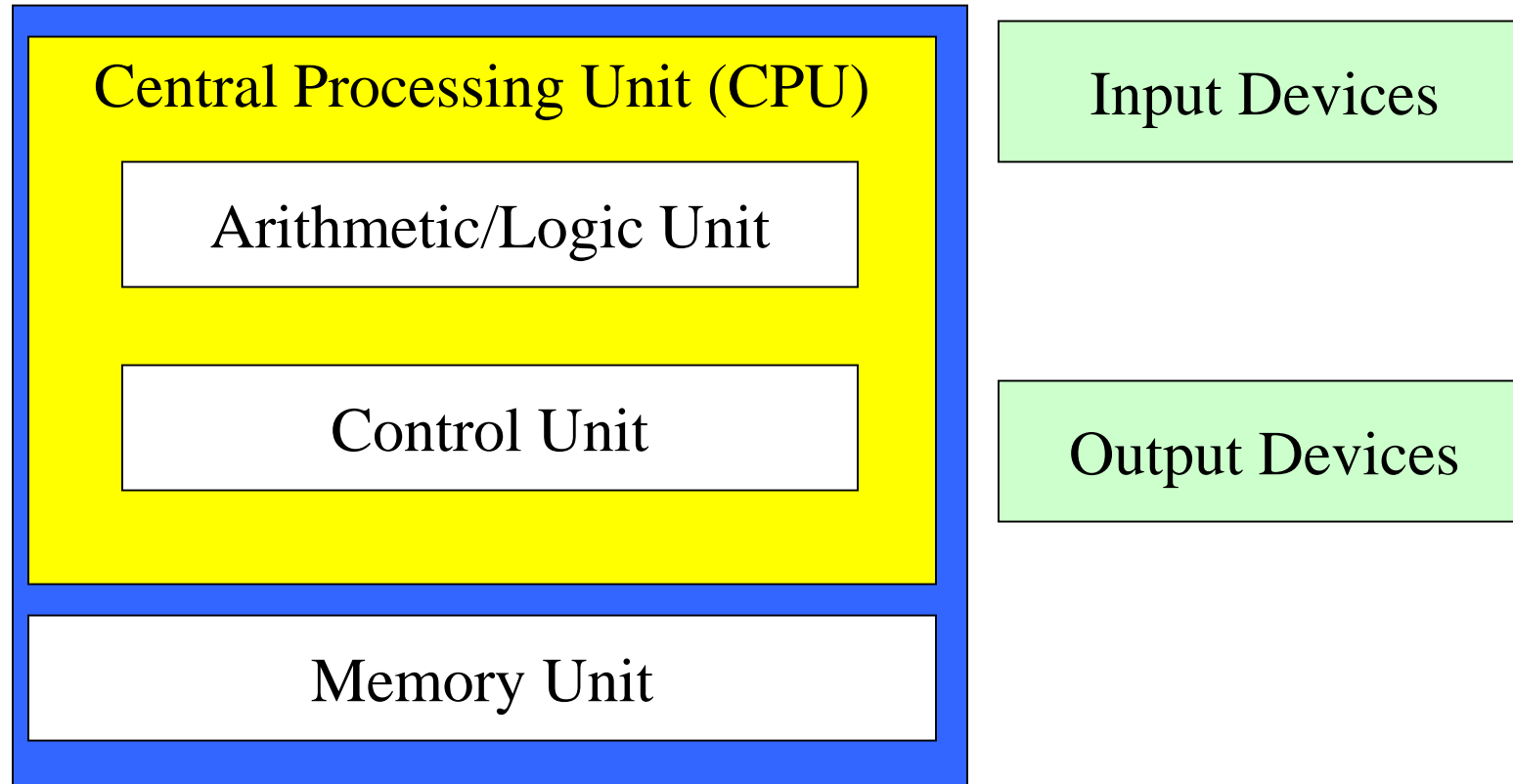
Computers cannot analyze problems and devise solutions.

Humans (that's us) must

- Analyze and understand a problem;
- Devise a sequence of steps to solve the problem;
- Translate the steps into a *computer language*.



Basic Computer Components



Central Processing Unit (CPU)

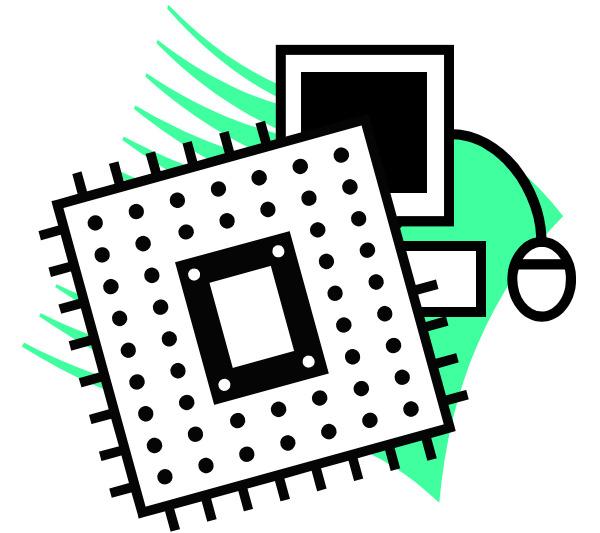
Executes stored *instructions*

Arithmetic/Logic Unit (ALU)

- Performs arithmetic and logical operations

Control Unit

- Controls the other components
- Guarantees instructions are executed in sequence



Memory Unit

Address

1001

1002

1003

1004

1005

1006

1007

1008

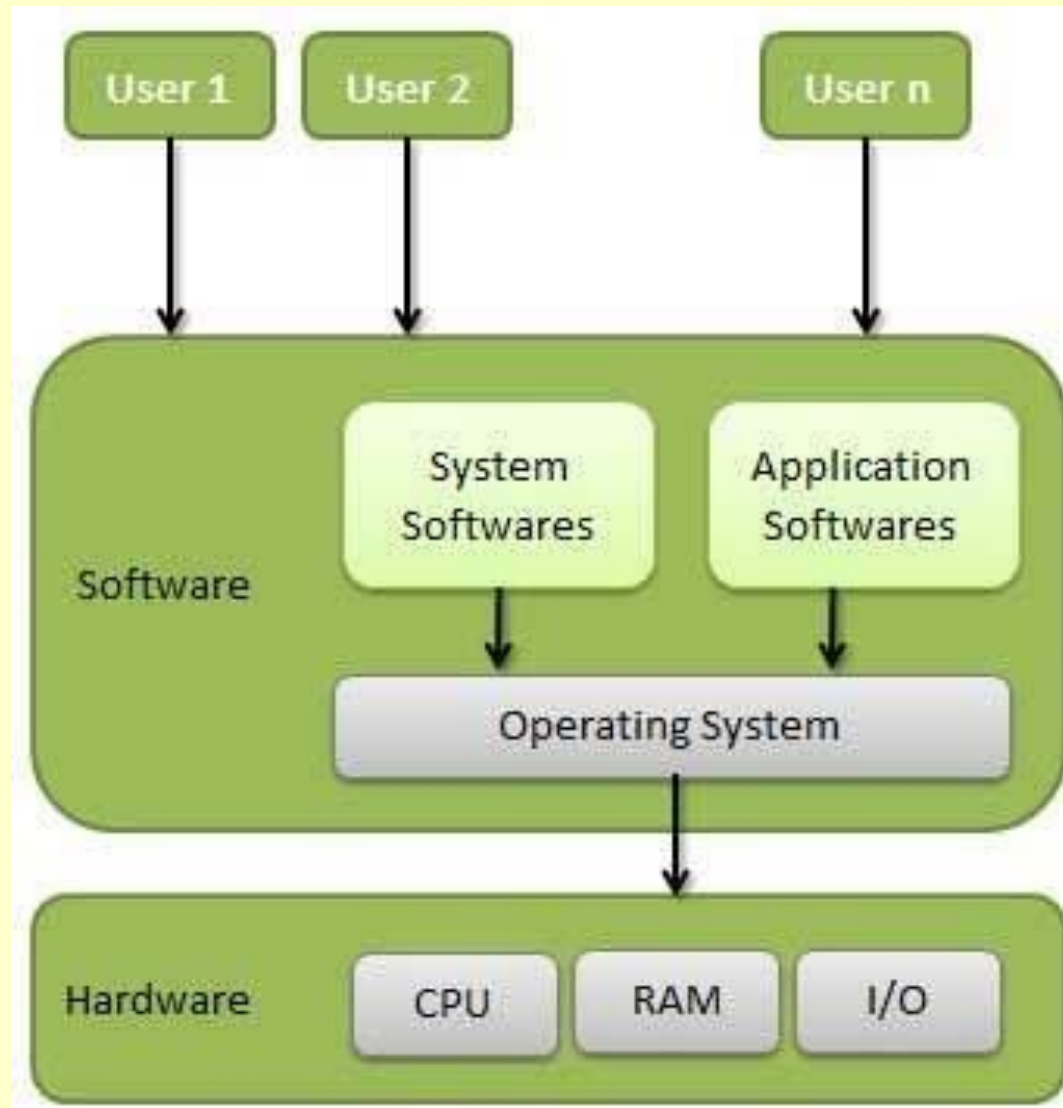
1009

Ordered sequence of *cells* or *locations*

Stores instructions and data in *binary*

Types of memory

- Read-Only Memory (ROM)
- Random Access Memory (RAM)



The 9 Most In-Demand Programming Languages

2017

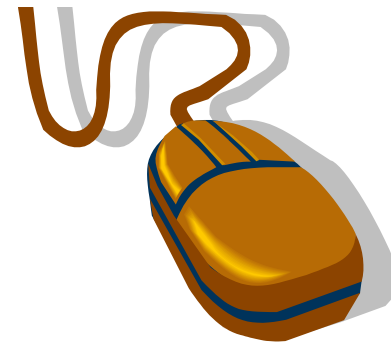
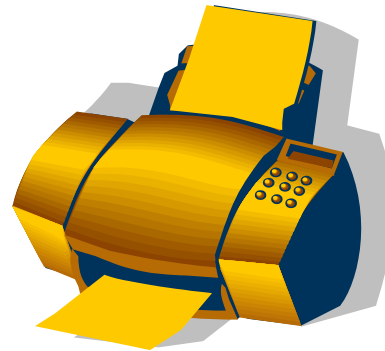
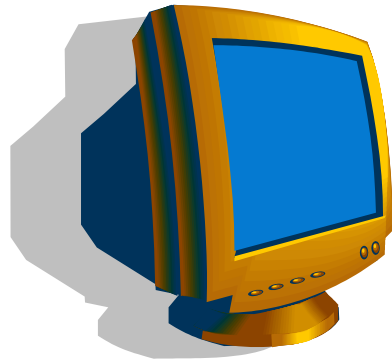
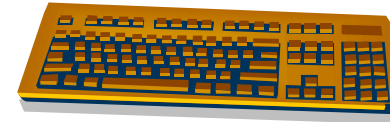


Input and Output Devices

Interaction with humans

Gathers data (Input)

Displays results (Output)



LECTURE 3

Introduction to Computer Programming



What is Computer Programming?

Planning or scheduling a sequence of steps for a computer to follow to perform a task.

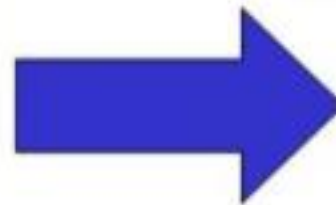
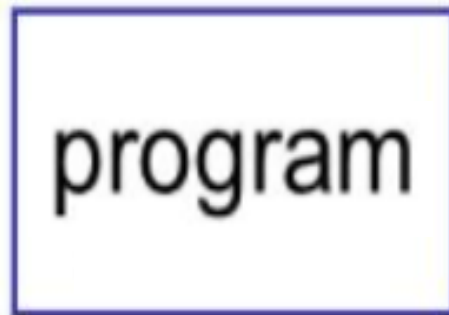
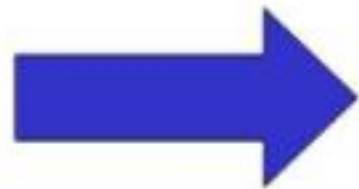
Basically, telling a computer what to do and how to do it.

What Is A Computer Program?

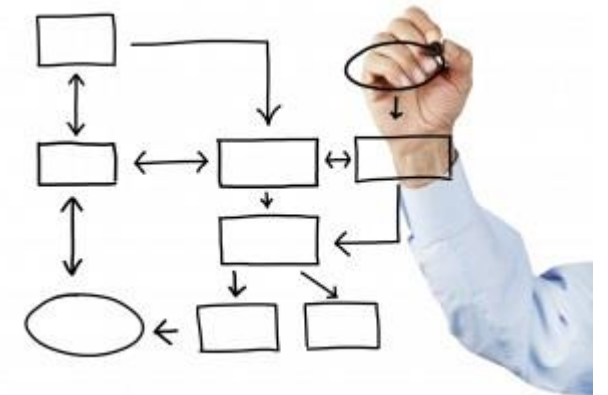
A sequence of steps to be performed by a computer.

Expressed in a **computer language**.

inputs



outputs



Program Flow Chart



A Computer Program In C++

```
// This program converts miles to kilometers.
// From Problem Solving, Abstraction, & Design Using C++
// by Frank L. Friedman and Elliot B. Koffman
#include <iostream>
using namespace std;
int main() {
    const float KM_PER_MILE = 1.609; // 1.609 km in a mile
    float miles,                    // input: distance in miles
           kms;                     // output: distance in kilometers
    // Get the distance in miles
    cout << "Enter the distance in miles: ";
    cin >> miles;
    // Convert the distance to kilometers and display it.
    kms = KM_PER_MILE * miles;
    cout << "The distance in kilometers is " << kms << endl;

    return 0;
}
```

LECTURE 4

Introduction to Programming Languages



Computer Languages

A set of

- Symbols (punctuation),
- Special words or keywords (vocabulary),
- And rules (grammar)

used to construct a computer program.



Differences In Computer Languages

Languages differ in

- Size (or complexity)
- Readability
- Expressivity (or writability)
- "Level"
 - closeness to instructions for the CPU

Machine Language

- Binary-coded instructions
- Used directly by the CPU
- Lowest level language
- Every program step is ultimately a machine language instruction

Address Contents

2034	10010110
2035	11101010
2036	00010010
2037	10101010
2038	10010110
2039	11101010
2040	11111111
2041	01010101
2042	10101101



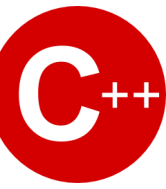
Assembly Language

- Each CPU instruction is labeled with a *mnemonic*.
- Very-low level language
 - Almost 1 to 1 correspondence with machine language
- Translated to machine language by an *assembler*.

Mnemonic	Instruction
ADD	10010011

Sample Program

```
MUL  X, 10
ADD  X, Y
STO  Z, 20
SUB  X, Z
```



High-Level Languages

- Closer to natural language
- Each step maps to several machine language instructions
- Provides support for *abstractions*
 - Easier to state and solve problems



Examples of High-Level Languages

Language	Primary Uses
Pascal	Learning to program
C++	General purpose
FORTRAN	Scientific programming
PERL	Web programming, text processing
Java	Web programming, application programming
COBOL	Business

Basic Programming Language Structures

Sequence

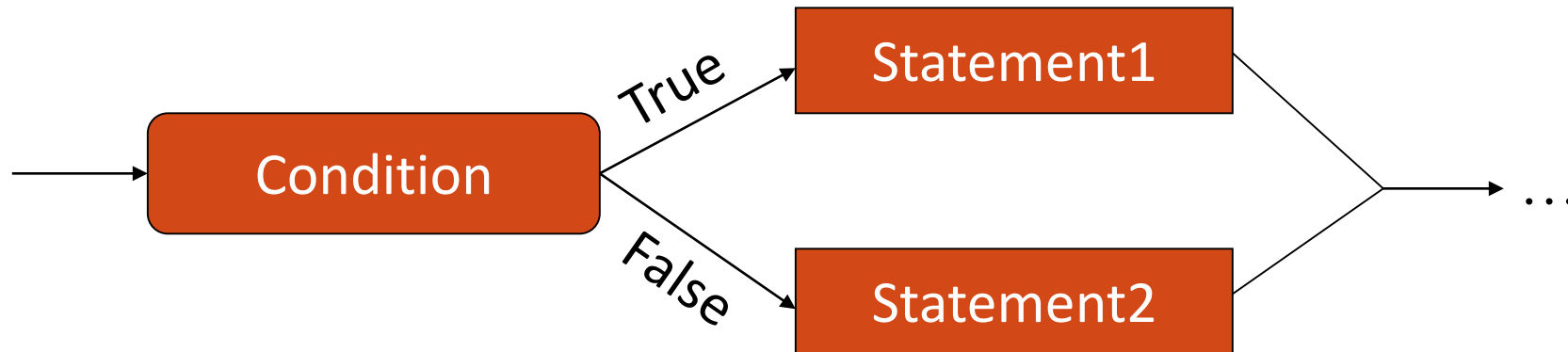
- Execute steps or *statements* in the language one after another.



Basic Programming Language Structures

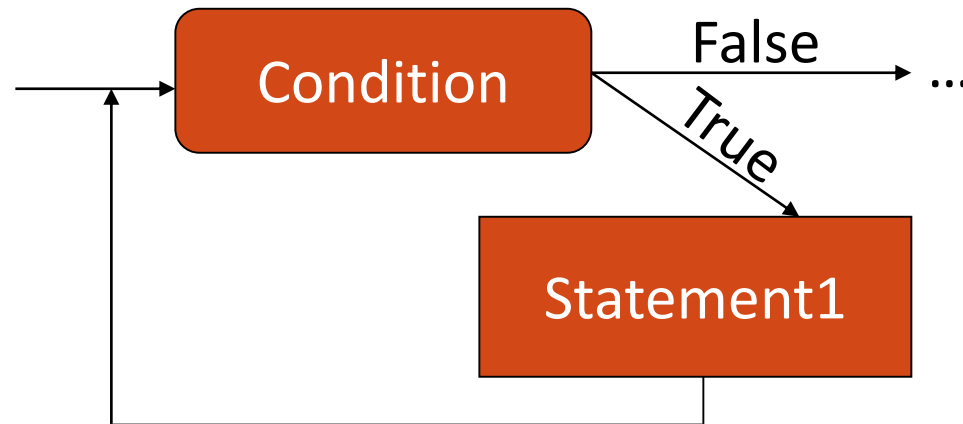
Selection (AKA *branch* or *decision*)

- Selectively execute statements based on some condition being true or false.
- IF condition THEN statement1 ELSE statement2



Basic Programming Language Structures

- Loop (AKA *repetition* or *iteration*)
 - Repeat a statement several times until a specified condition is false.
 - `WHILE condition DO statement1`



Basic Programming Language Structures

Subprogram (AKA *procedure*, *function*, *method*, or *subroutine*)

- A collection of the previous structures that accomplishes some smaller task.

