Introductory Class

Operating System (CSC1036 & INF1036)

Dr. Hasin A. Ahmed
Assistant Professor
Department of Computer Science
Gauhati University

Course Content

- **UNIT-I: Review of computer organization**
- **UNIT-II: Memory architecture**
- **UNIT-III: Support for concurrent process**
- **UNIT-IV: Scheduling**
- **UNIT-V: System deadlock**
- **UNIT-VI:Multiprogramming System**
- **UNIT-VII: Advanced Topics**

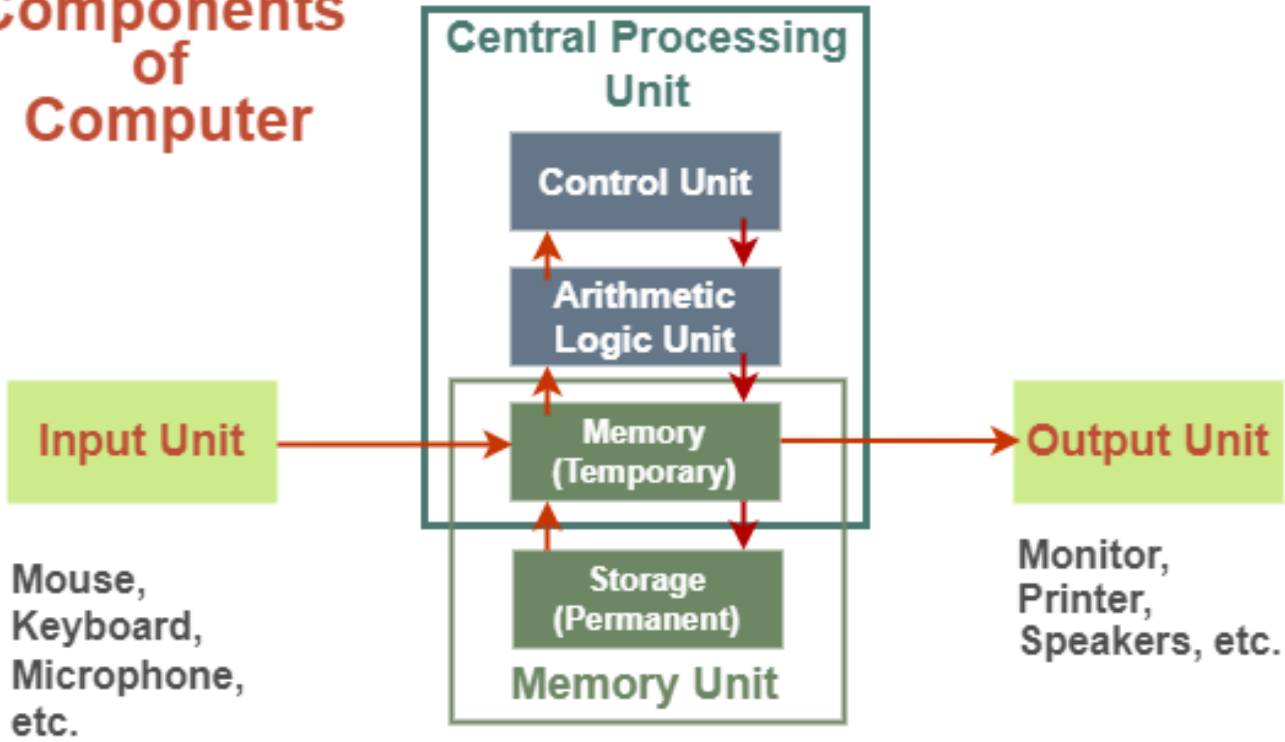
Evaluation Process

- **Class Test I: 20**
- **Class Test II:20**
- **Seminar: 5**
- **Laboratory: 15**
- **End term: 40 (To be scaled down from 100)**

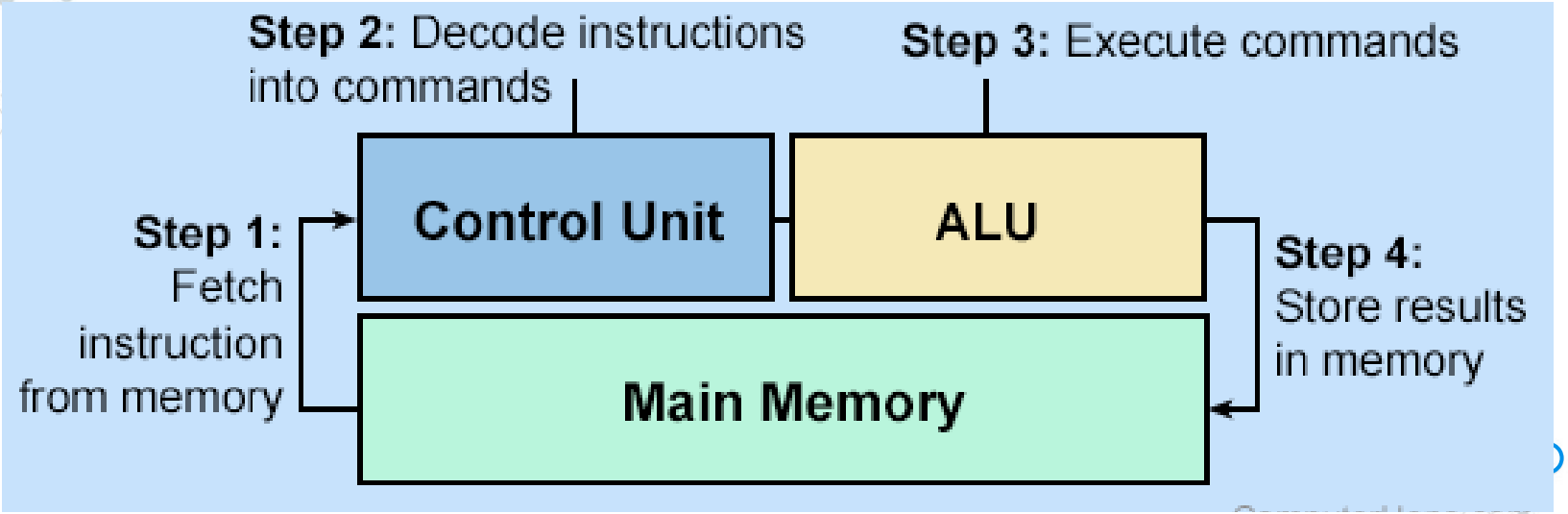
Laboratory Component

1. Basic UNIX commands
2. Shell Programming
3. Write programs using the following system calls of UNIX operating system fork, exec, getpid, exit, wait, close, stat, opendir, readdir etc.
4. Windows API tutorial
5. Basic Windows commands
6. Batch scripting in windows

Components of Computer



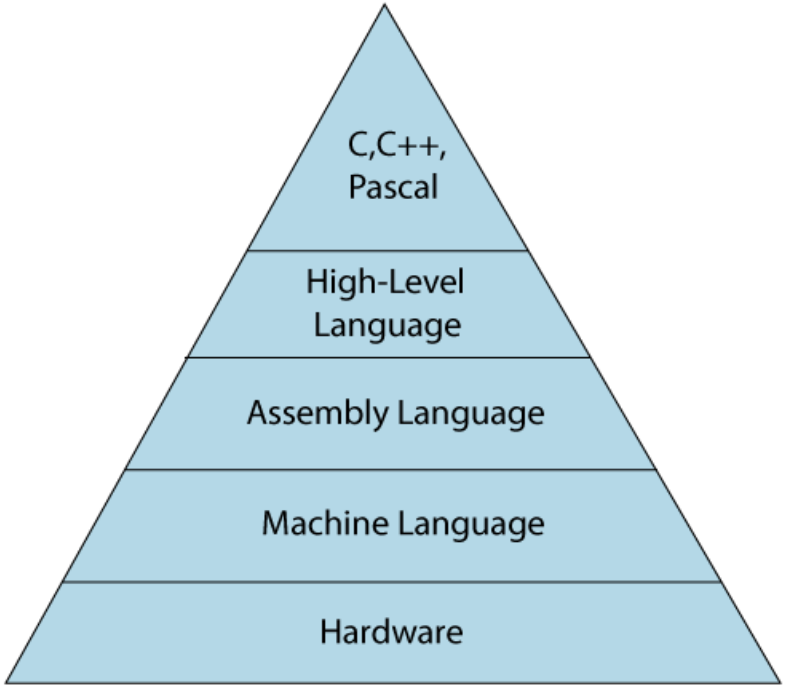
Program


Step 2: Decode instructions into commands
Step 3: Execute commands
Control Unit
ALU
Step 1: Fetch instruction from memory
Step 4: Store results in memory
Main Memory

Programming languages

Translators

Types of programming languages



Machine level language

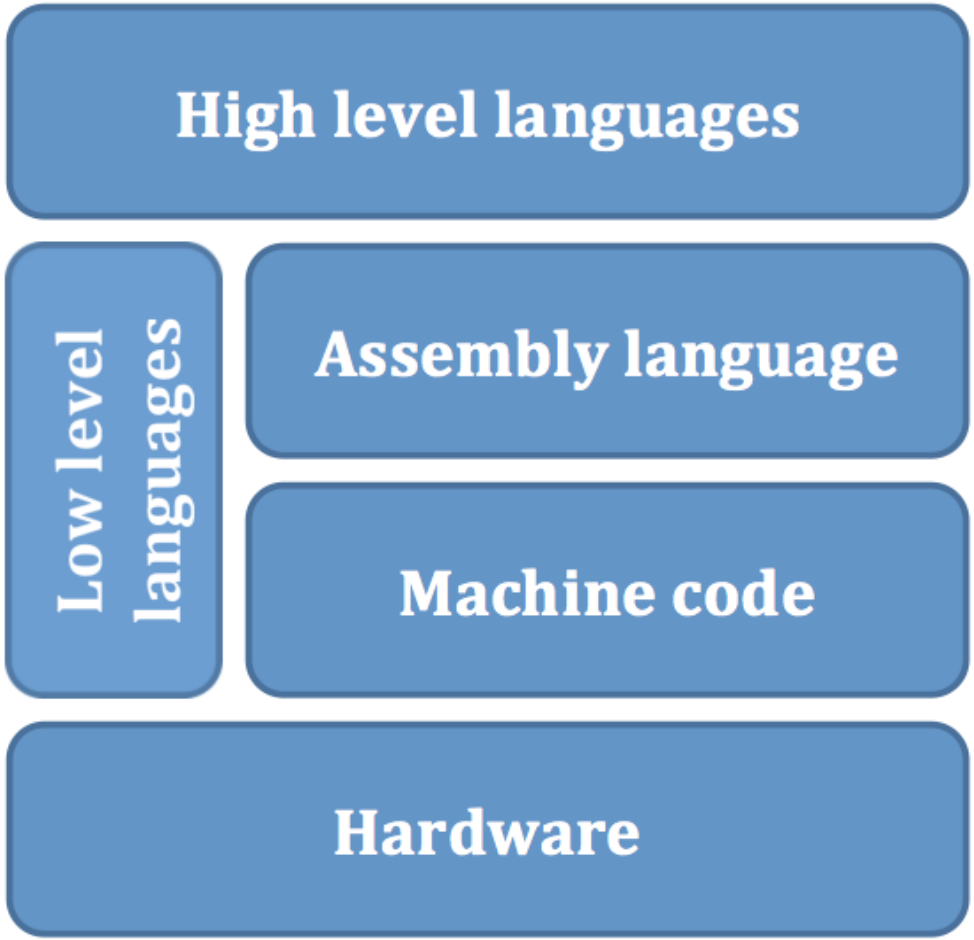
- The machine-level language is a language that consists of a set of instructions that are in the binary form 0 or 1
- As we know that computers can understand only machine instructions
- Creating a program in a machine-level language is a very difficult task

Assembly language

- In assembly language, binary codes are replace by alphanumeric mnemonics
- There is a one to one correspondence between assembly code and machine code
- Assembler converts assembly language to its machine level language equivalent

High level language

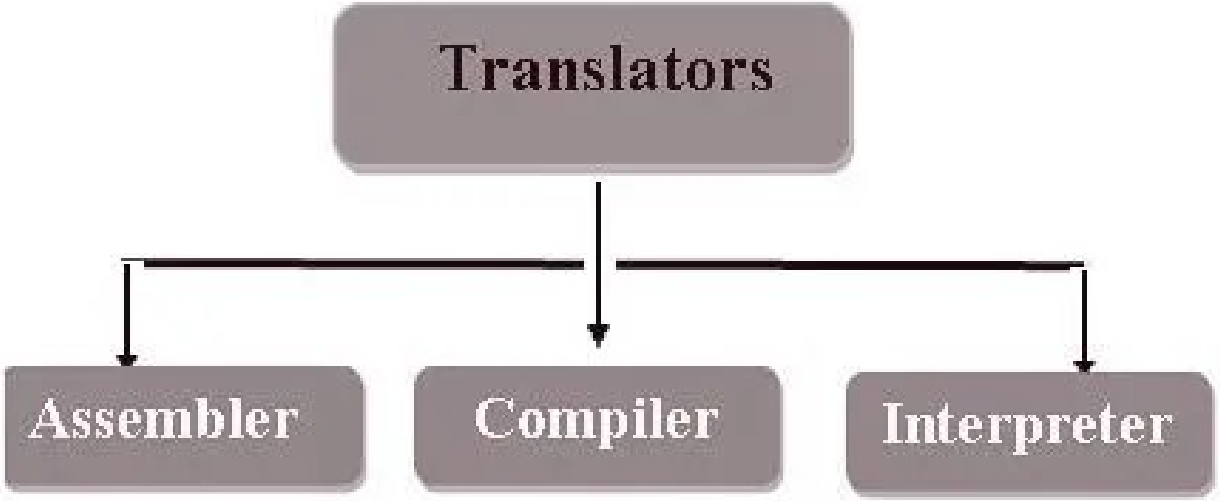
- In High level language, codes are written in a form very similar to natural language
- One high level instruction usually leads to multiple machine instruction
- Compiler and Interpreter converts high level language to machine level language


High level languages
Low level languages
Assembly language
Machine code
Hardware

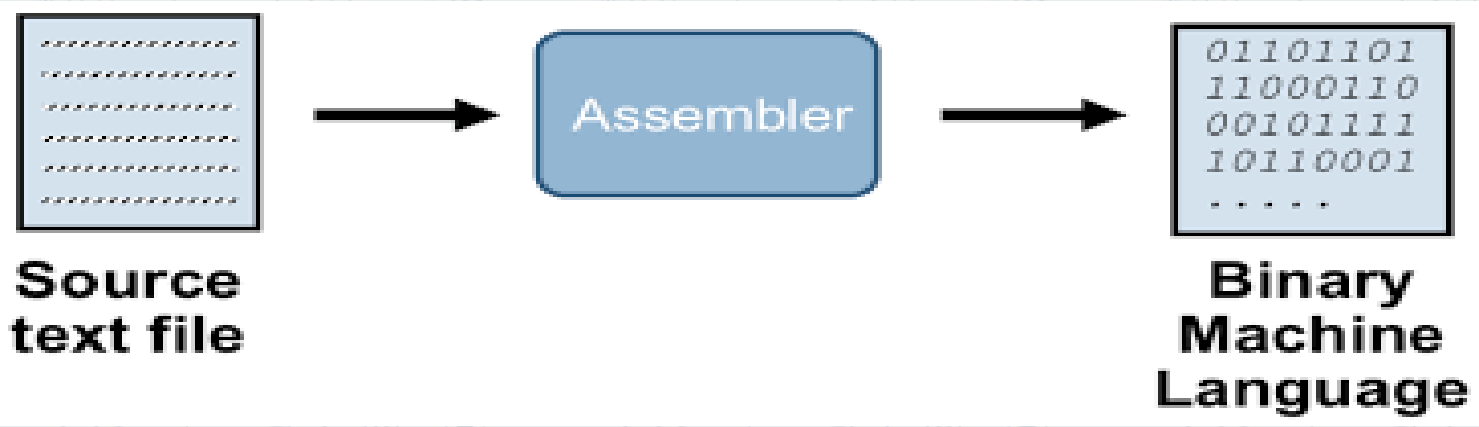
High level languages

High-Level Language	Low-level language
It can be considered as a programmer-friendly language.	It is considered as a machine-friendly language.
It requires a compiler/interpreter to be translated into machine code.	It requires an assembler that would translate instructions.
It can be ported from one location to another.	It is not portable.
It is easy to understand.	It is difficult to understand.
It is easy to debug.	It is difficult to debug.
It is less memory efficient, i.e., it consumes more memory in comparison to low-level languages.	It consumes less memory.

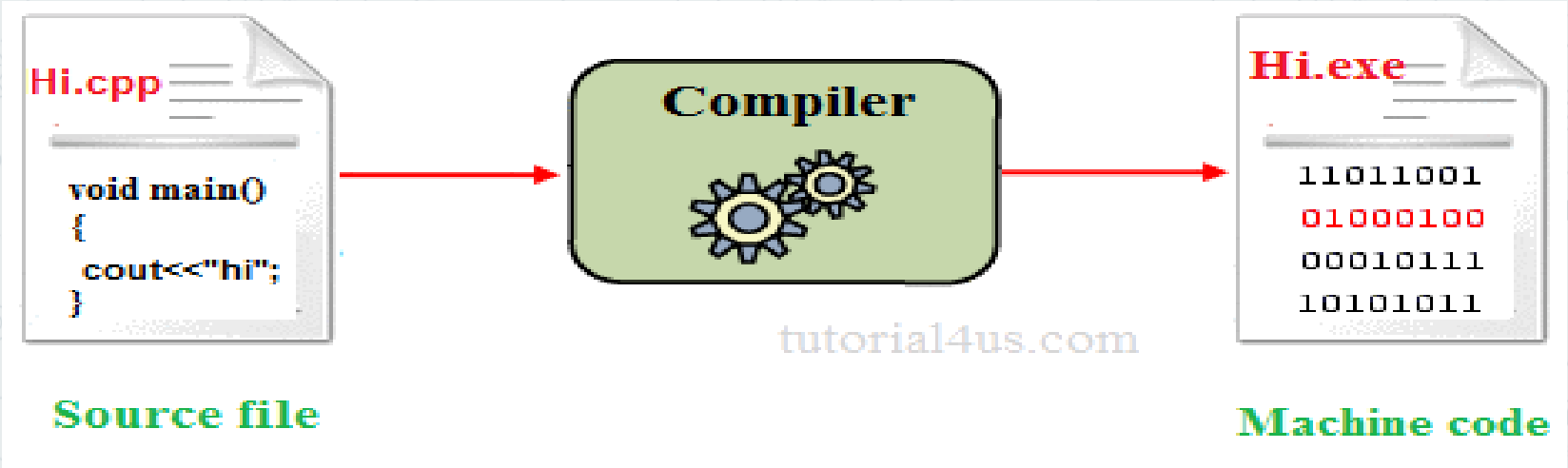
Types of translators



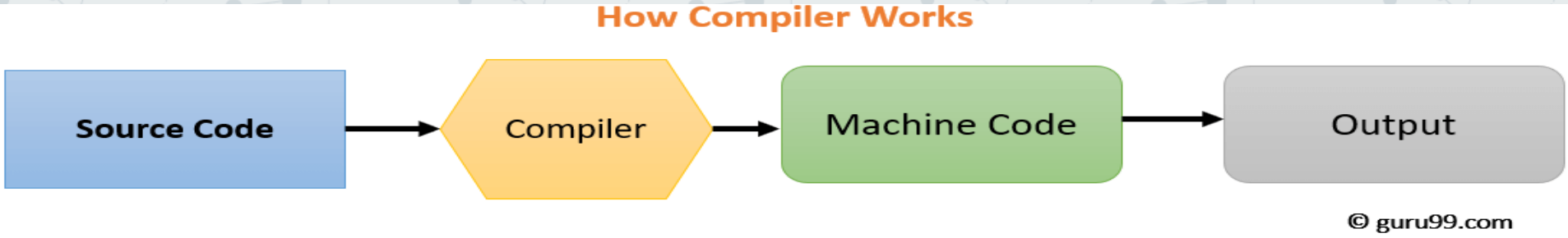
Assembler

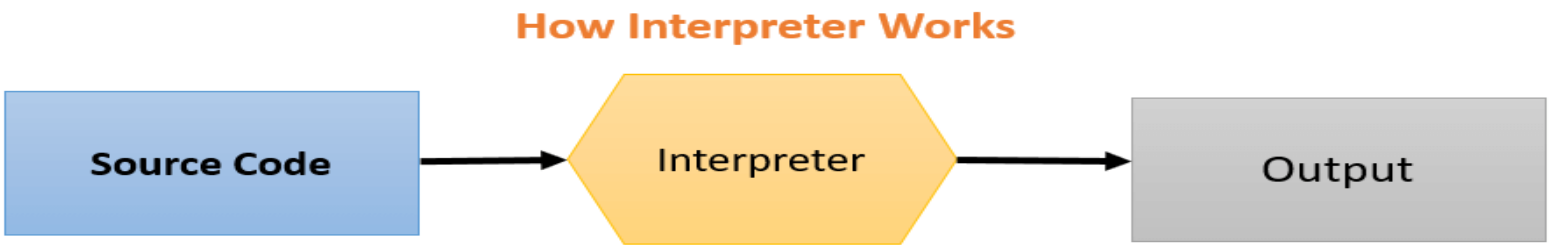

Assembler
01101101
11000110
00101111
10110001
. . . . .
Source text file
Binary Machine Language

Compiler


Hi.cpp
void main()
{
cout<<"hi";
}
Compiler
Hi.exe
11011001
01000100
00010111
10101011
tutorial4us.com
Source file
Machine code

Interpreter





Compilers vs Interpreters

- Interpreters are slower than compilers when the program is running because they have to interpret each line
- The interpreter needs to be in memory each time the program is run, whereas compilers do not
- Interpreters can make debugging a program easier because they can track errors line by line
- Compilers have to have programs recompiled each time the code is changed, even if the change is minor

Software

- Software is a set of programs used to operate computers and execute specific tasks
- Opposite of hardware, which describes the physical aspects of a computer, software is a generic term used to refer to scripts and programs that run on a device
- Software: Application software and System software

Application software

- Application software (app for short) is a program or group of programs designed for a specific purpose from end users' perspective
- Examples are web browser, word processors, spreadsheet software

System software

- System Software is a set of programs that control and manage the operations of computer hardware
- It also provides an environment for development ad execution of application

System Software Vs Application Software

Vs

Operating System

- **An operating system (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs**

thank you

? ? ?