Overview of Computers and Programming

Mirza Mohammad Lutfe Elahi

Outline

- Overview of Computers
  - Hardware
  - Software
- Computer Languages
- Software Development Method
- Pseudo Code and Flowcharts
- Professional Ethics

Computers

- Computers receive input, store, process, and output information.
- Computer can deal with numbers, text, images, graphics, and sound.
- Computers are worthless without programming.
- Programming Languages allow us to write programs that tell the computer what to do and to provide a way to communicate with computers.
- Programs are then converted to machine instructions so the computer can understand it.

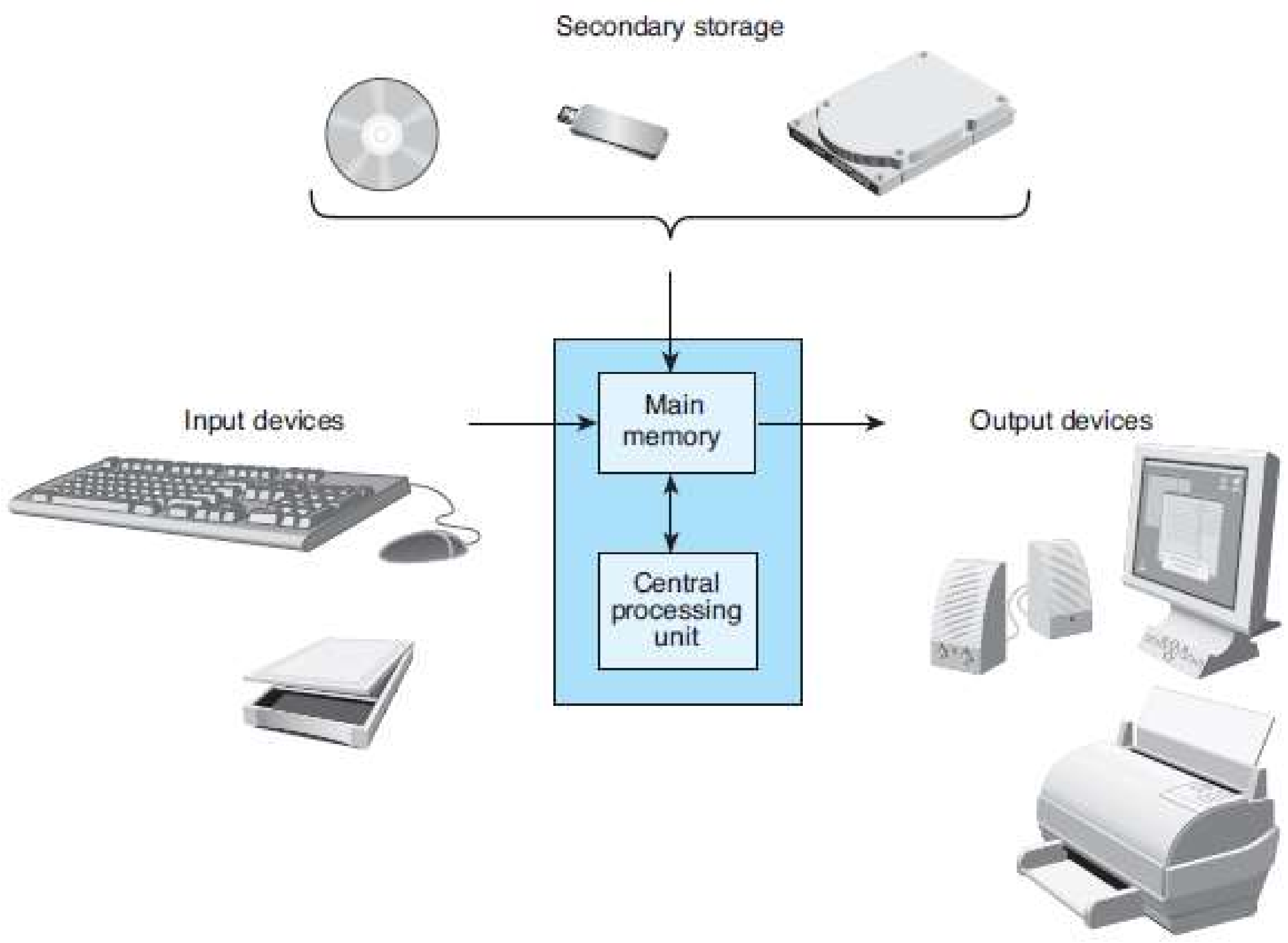
Hardware & Software

- Hardware is the equipment used to perform the necessary computations.
  - Central Processing Unit (CPU), memory, disk storage, monitor, keyboard, mouse, printer, etc.
- Software consists of the programs that enable us to solve problems with a computer by providing it with a list of instructions to follow
  - Windows OS, MS Word, Mozilla Firefox, etc.

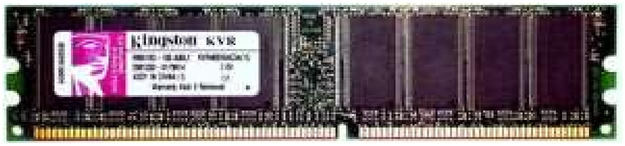
Computer Hardware

- **Main Memory**
  - **RAM** - Random Access Memory - Memory that can be read and written in any order (as opposed to sequential access memory), volatile.
  - **ROM** - Read Only Memory - Memory that cannot be written to, non-volatile.
- **Secondary Memory:** Magnetic hard disks, Flash (solid state) disks, Optical disks (CDs and DVDs).
- **Central Processing Unit (CPU):** Executes all computer operations and perform arithmetic and logical operations.
- **Input/Output Devices:** keyboard, mouse, scanner, monitor, printer, and speakers.
- **Computer Networks** – Computers that are linked together can communicate with each other.

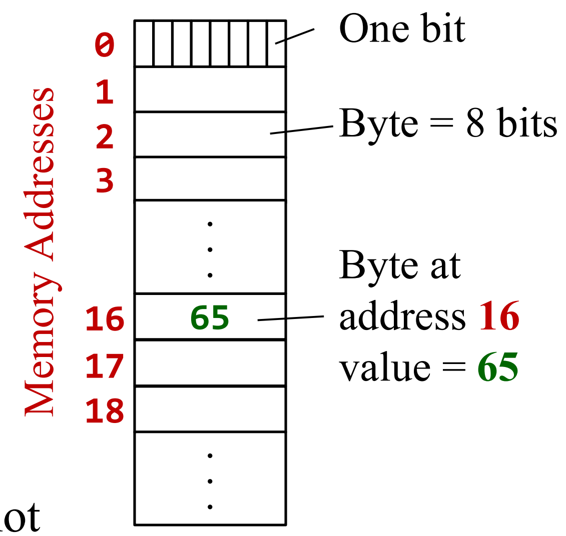
Components of a Computer



Memory



- Memory: a large collection of memory cells
- Each Memory Cell has an **address** and a **value**
- Bit: Binary digit = Either 0 or 1
- Byte: Made up of 8 bits
- Memory Address: position of a memory cell
- Memory Content: Value stored in memory
  - Every memory cell has content, whether we know it or not
- Memory capacity
  - Kilobyte (KB) = 2^{10} = 1024 Bytes; Megabyte (MB) = 2^{20} Bytes > 10^6 Bytes
  - Gigabyte (GB) = 2^{30} > 10^9 Bytes; Terabyte (TB) = 2^{40} Bytes > 10^{12} Bytes



Computer Software

- Operating System - controls the interaction between machine and user. Examples: Windows, Linux, etc.
  - Communicates with computer user.
  - Collects input and Displays output.
  - Manages memory and processor time.
  - Manages Storage Disk.
- Application Software - developed to assist a computer user in accomplishing specific tasks. Example: MS Word, Google Chrome, etc.

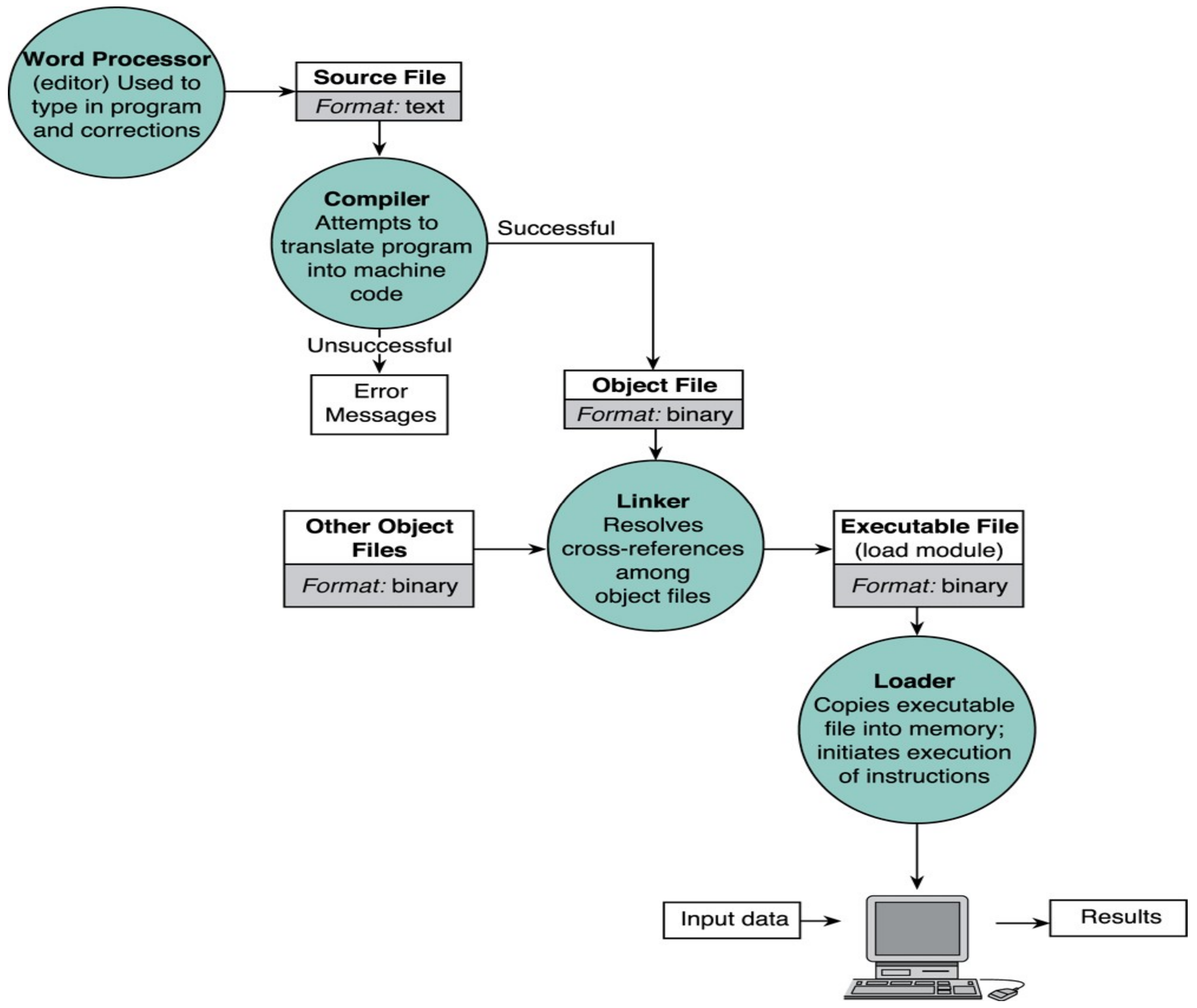
Computer languages

- **High-level Language:** Combines algebraic expressions and high-level commands
  - High Level **:** Very far away from the actual machine language
  - Examples: Fortran, C, Prolog, C#, Perl, and Java.
- **Machine Language:** A collection of machine instructions
  - Not standardized. There is a different machine language for every processor family.
- **Assembly Language:** uses symbols (called mnemonics) that correspond to machine language instructions.
  - Low level: Very close to the actual machine language.

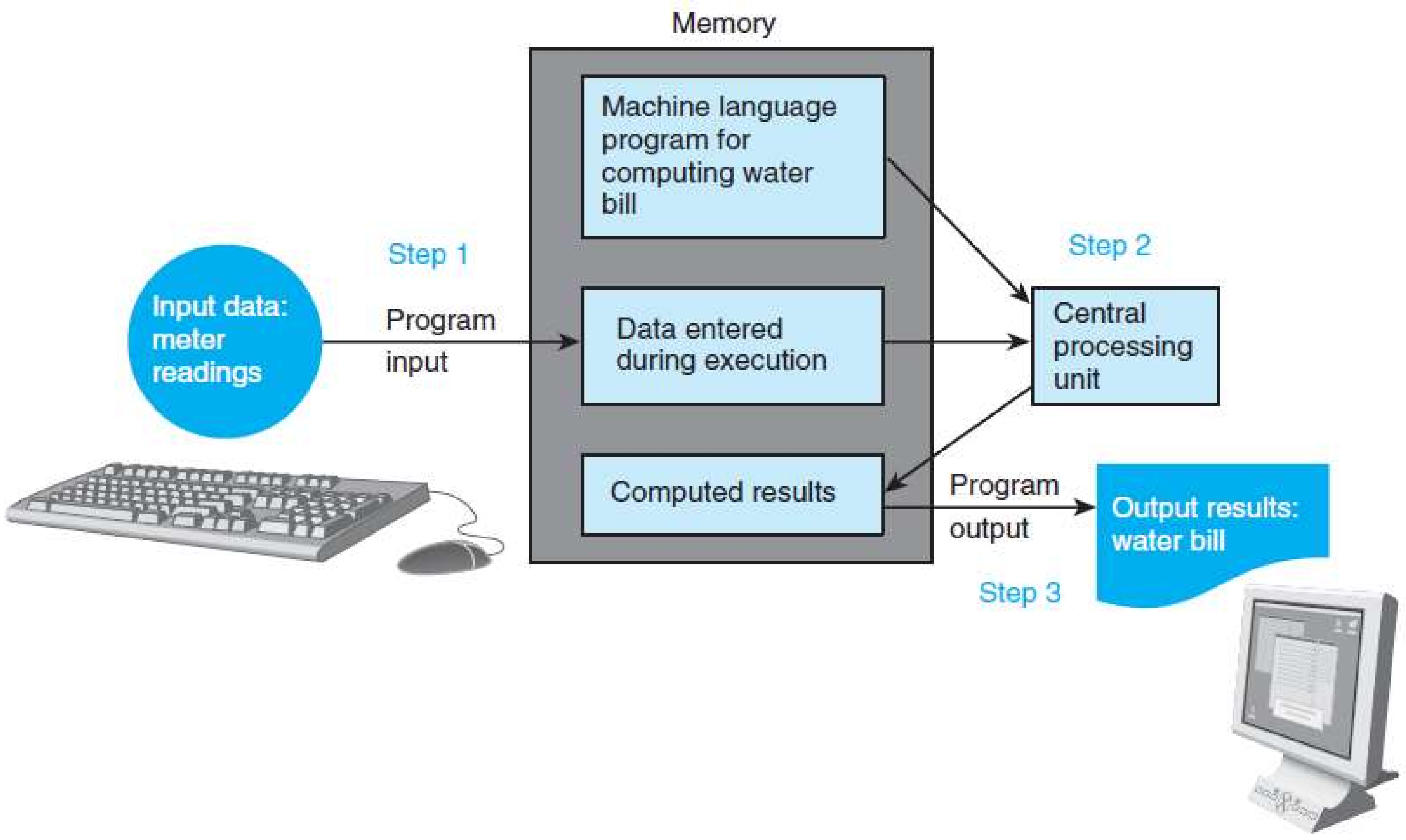
Compiler

- Compilation is the process of translating the source code (high-level) into executable code (machine level).
- **Source file:** contains the original program code
  - A Compiler turns the Source File into an Object File
- **Object file:** contains machine language instructions
  - A Linker turns the Object File into an Executable
- **Integrated Development Environment (IDE):** a program that combines simple text editor with a compiler, linker, loader, and debugger tool
  - Examples: Code::Blocks, Eclipse, Visual Studio, etc.

Editing, Translating, Linking, and Running High-Level Language Programs


Word Processor
(editor) Used to type in program and corrections
Source File
Format: text
Compiler
Attempts to translate program into machine code
Successful
Unsuccessful
Error Messages
Object File
Format: binary
Other Object Files
Format: binary
Linker
Resolves cross-references among object files
Executable File
(load module)
Format: binary
Loader
Copies executable file into memory; initiates execution of instructions
Input data
Results

Flow of Information During Program Execution


Memory
Machine language program for computing water bill
Step 1
Input data: meter readings
Program input
Data entered during execution
Step 2
Central processing unit
Computed results
Program output
Output results: water bill
Step 3

Software Development Method

1. **Specify** problem requirements
2. **Analyze** the problem
3. **Design** the algorithm to solve the problem
4. **Implement** the algorithm
5. **Test** and **verify** the completed program
6. **Maintain** and **update** the program

Steps Defined

1. **Problem:** statement that specifies the problem that should be solved on the computer.
2. **Analysis**: Understanding the problem and identifying the inputs, outputs, and required computation.
3. **Design** - Designing and developing the list of steps called **algorithm** to solve the problem.
4. **Implementation:** writing the algorithm as a program using a given programming language.
5. **Testing** - Testing requires checking and verifying that the program actually works as desired.
6. **Maintenance** - Maintaining involves finding previously undetected errors and keep it up-to-date.

Converting Miles to Kilometers

1. **Problem**: Your boss wants you to convert a list of miles to kilometers. Since you like programming, you decide to write a program to do the job.
2. **Analysis**
   - We need to receive miles as input
   - We need to output kilometers
   - We know 1 mile = 1.609 kilometers
3. **Design**
   1. Get distance in miles
   2. Convert to kilometers
   3. Display kilometers

Implementation in C Language

```c
/*
 * Converts distance in miles to kilometers.
 */
#include <stdio.h>           // printf, scanf definitions
#define KMS_PER_MILE 1.609   // conversion constant

int main(void) {
   float miles,   // input - distance in miles
         kms;     // output - distance in kilometers

   /* Get the distance in miles */
   printf("Enter the distance in miles> ");
   scanf("%f", &miles);

   /* Convert the distance to kilometers */
   kms = KMS_PER_MILE * miles;

   /* Display the distance in kilometers */
   printf("That equals %f kilometers.\n", kms);

   return 0;
}
```

```
Sample Run:
Enter the distance in miles> 10.0
That equals 16.090000 kilometers
```

Converting Miles to Kilometers

5. Test: We need to test the previous program to make sure it works. To test we run our program and enter different values and make sure the output is correct.

6. Maintenance: Next time, your boss wants to add a new feature, so he wants you to add support for converting different units.

Pseudo Code and Flowchart

- **Algorithm** - A list of steps for solving a problem.
- **Pseudo code** - A combination of English phrases and language constructs to describe the algorithm steps.
- **Flowchart** - A diagram that shows the step-by-step execution of a program

Why Use Pseudo Code?

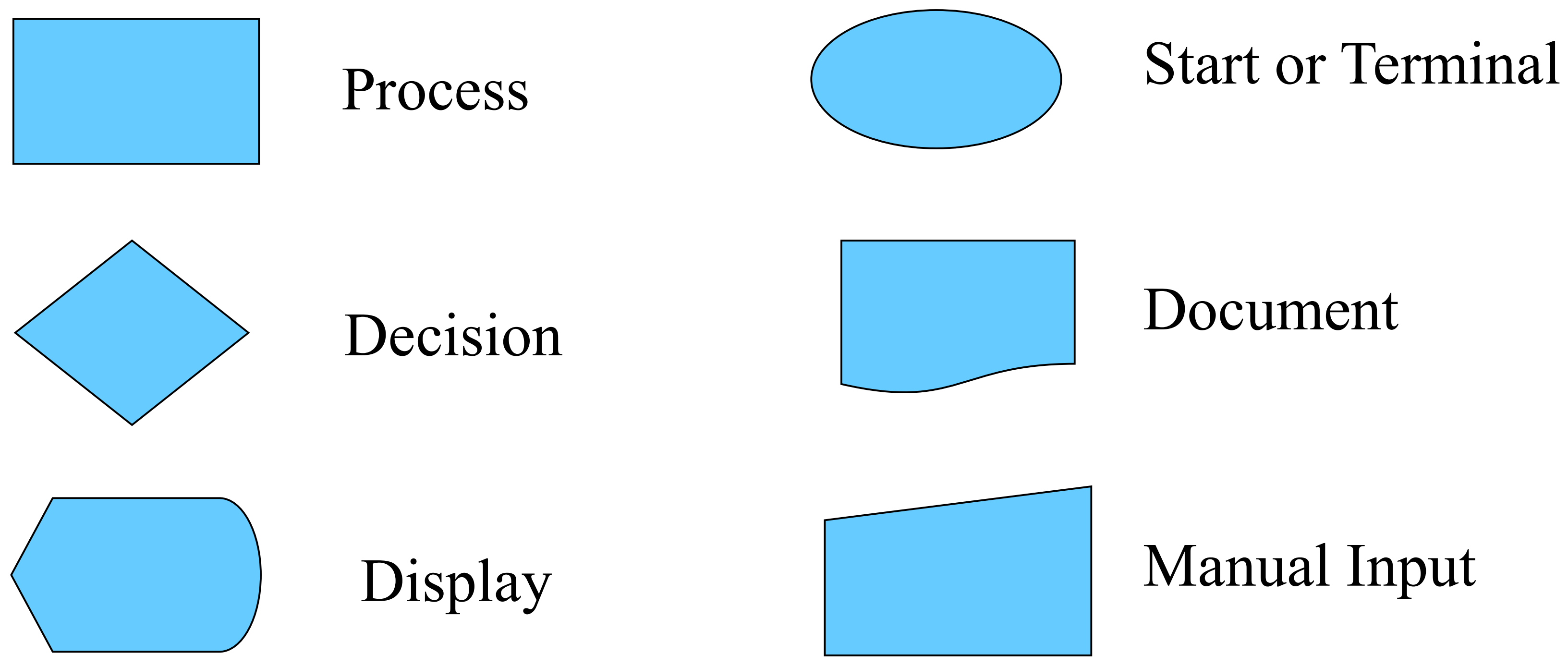
- The benefit of pseudo code is that it enables the programmer to concentrate on the algorithm without worrying about all the syntactic details of a particular programming language.
- In fact, you can write pseudo code without even knowing what programming language you will use for the final implementation.
- Pseudo code cannot be compiled or executed, and does not follow syntax rules. It is simply an important step in producing the final code.
- Example:

```
Input Miles
Kilometers = Miles * 1.609
Output Kilometers
```

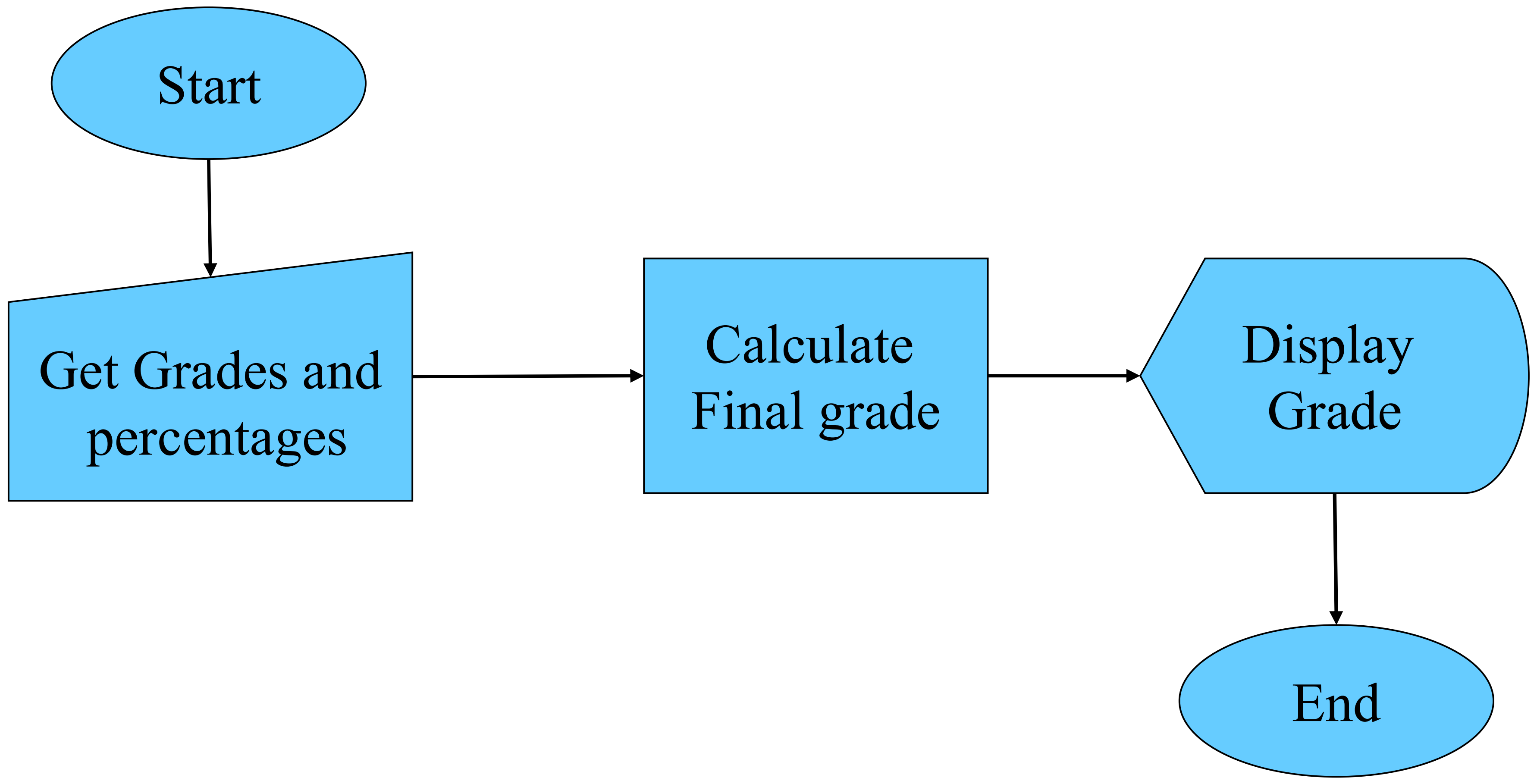
Another Example of Pseudo Code?

- **Problem**: Calculate your final grade for CSE 115
- **Specify the problem:** Get different grades and then compute the final grade.
- **Analyze the problem:** We need to input grades for quizzes, assignments, exams, class performance and the percentage each part counts for. Then we need to output the final grade.
- **Design**
  1. Get the grades: exams, quizzes, assignments, and labs.
  2. Grade = 0.2 * Quizzes + 0.1 * Assignments + 0.25 * Midterm Exam + 0.4 * Final Exam + 0.05 * Class Performance
  3. Output the Grade
- **Implement and Test:** Learn how to program in C, Write the program, then input some test values, calculate and check the final grade.

Flowchart


Process
Start or Terminal
Decision
Document
Display
Manual Input

Example of Flowchart



Professional Ethics

- **Privacy and Misuse of Data**
  - **computer theft (computer fraud) -** Illegally obtaining money by falsifying information in a computer database

- **Computer Hacking**
  - **Virus -** Code attached to another program that spreads through a computer's disk memory, disrupting the computer or erasing information
  - **Worm -** A virus that can disrupt a network by replicating itself on other network computers

Professional Ethics

- **Plagiarism and Software Piracy**
  - **Software piracy –** Violating copyright agreements by illegally copying software for use in another computer

- **Misuse of a Computer Resource**