

Lecture 1

CMP1201: COMPUTER PROGRAMMING FUNDAMENTALS

Dr. Andrew Katumba



Syllabus

- Two lectures a week: Tuesday and Thursday at 8:00 am
- 2 written tests, 1 practical test, weekly assignments
- This is a hands-on course. You MUST Code. Code. Code.
- Work in groups of 5 for group assignments
- If necessary, a cloud LINUX server will be made available to run your code. Sign up with your class representatives if interested.

Course website

Muele.mak.ac.ug

Key: 120122

Course Books

- 1) C Programming Language by Brian Kernighan and Dennis Ritchie, 2nd Edition (CPL)
- 2) Problem Solving and Program Design in C by Jeri R. Hanly, Elliot B. Koffman, 8th Edition, Global Edition (PSC)

IDE

- Visual Studio Code (all platforms)
- Visual Studio Community (<https://visualstudio.microsoft.com/vs/community/>),
Windows only

What you will learn

- Fundamental Programming Concepts
- Key techniques
- Basic C/C++ facilities

By the end of the course you will be able to

- Write small C programs
- Read much larger programs
- Learn the basics of many other languages
- Proceed to advanced C/C++ courses

Academic Integrity

Discuss assignments with your classmates, but you **MUST** write all your code and all your answers yourself

How to succeed

Ask questions

- In class
- In office hours, tutorials/labs

Study together and discuss assignments with each other (without plagiarizing!)

Textbook

- Read and re-read the material
- Complete practice problems

Start coding and studying early

Instructor

- **Email:** andrew.katumba@mak.ac.ug
- **Tel:** +256 784 343 505
- **Office:** 3002, CEDAT new building
- **Office hours (open consultation):** Friday 11am - 12pm

TAs:

- Ben Wycliff Mugalu (ben12wycliff@gmail.com)
- Alvin Mulumba(Head TA, michaelmulumbaalvin@gmail.com)

Computer Programs

- A program provides a computer with a set of simple instructions to achieve a goal
- “Programming is how you get computers to solve problems.”
- With the right skills and experience, a programmer can craft software to solve an unlimited number of problems

Programs are everywhere

On your computer:

- Web browser
Request and display information from distant sites
- Word processor
Record text, change appearance, save to disk
- Music player
Organize mp3's, select time in song, play, stop

Programs are everywhere

In the restaurant:

- Cashier

Compute price of food purchase, charge payment to account, (if pay cash: compute change)

- HVAC

Monitor temperature, adjust A/C or heating

- Electronic signs

Display menus and prices, load and display university news

Programs are everywhere

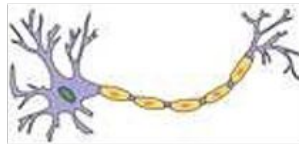
In humans:

- Sports

When to run, where to run; when to pass; who to pass to; when to shoot

- The brain

Neurons working together to combine information about an image to recognize a dog or a cat



dog

Programming and Problem solving

A sequence of precise instructions which leads to a solution is called an **algorithm**.

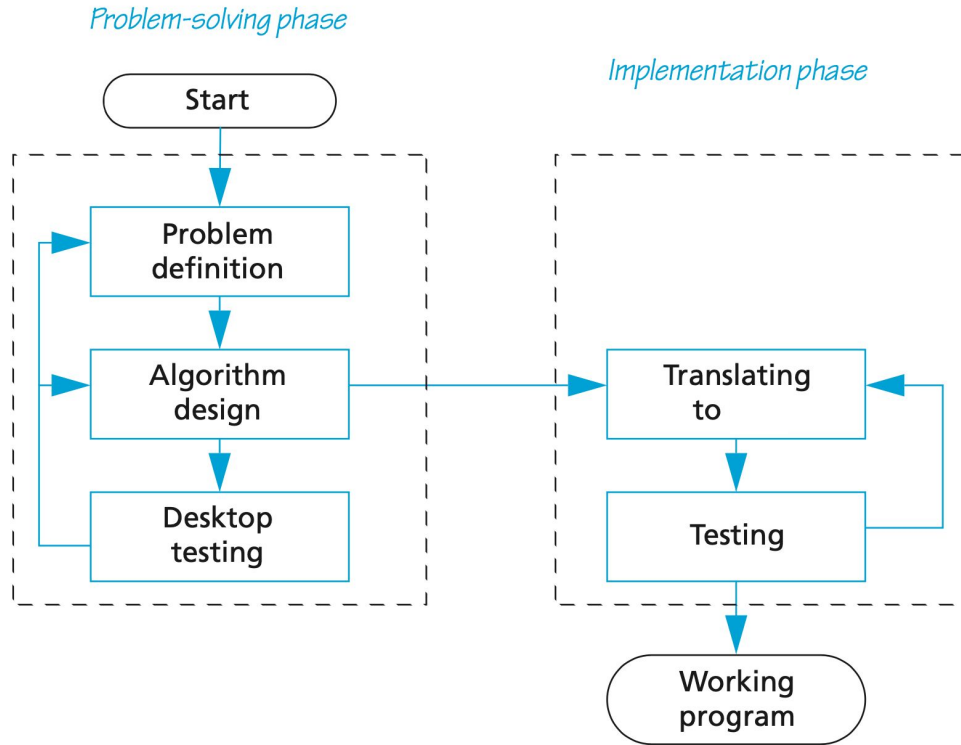
Some approximately equivalent words are **recipe**, **method**,

Algorithm Example

Algorithm that determines how many times a name occurs in a list of names:

1. Get the list of names.
2. Get the name being checked.
3. Set a counter to zero.
4. Do the following for each name on the list:
Compare the name on the list to the name being checked,
and if the names are the same, then add one to the counter.
5. Announce that the answer is the number indicated by the counter.

Program Design process



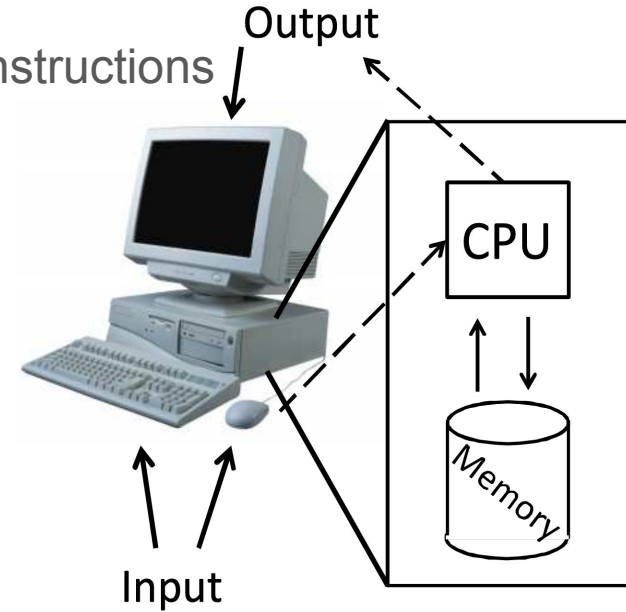
Computer System Structure

The Central Processing Unit (CPU) - performs all the instructions

Memory - stores data and instructions for the the CPU

Input - collects information from the world

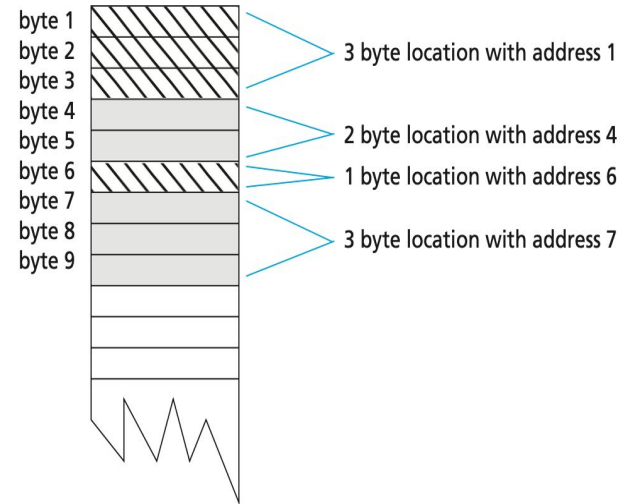
Output - provides information for the world



Computer Memory

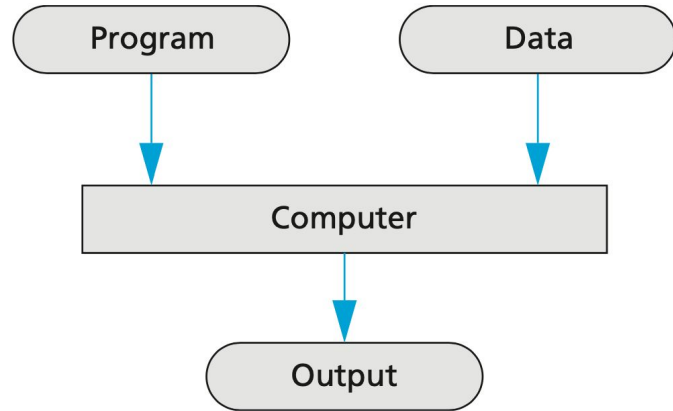
Bytes and Addresses

Main memory is divided into numbered locations called **bytes**. The number associated with a byte is called its **address**. A group of consecutive bytes is used as the location for a data item, such as a number or letter. The address of the first byte in the group is used as the address of this larger memory location.



Software

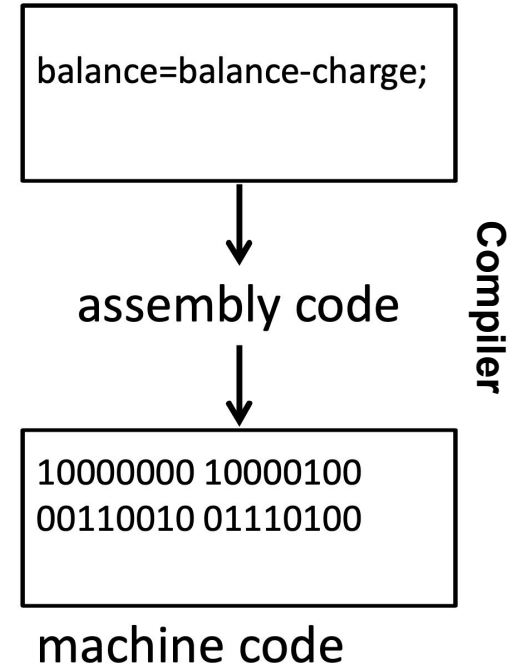
Software is a set of programs that enables the hardware to perform a specific task.



A running program

C- high-level language

- High level language
 - Uses words to describe instructions
 - More intuitive to people
 - Computer-independent
- Machine language
 - Uses binary to describe instructions
 - Less intuitive to people
 - Computer dependent

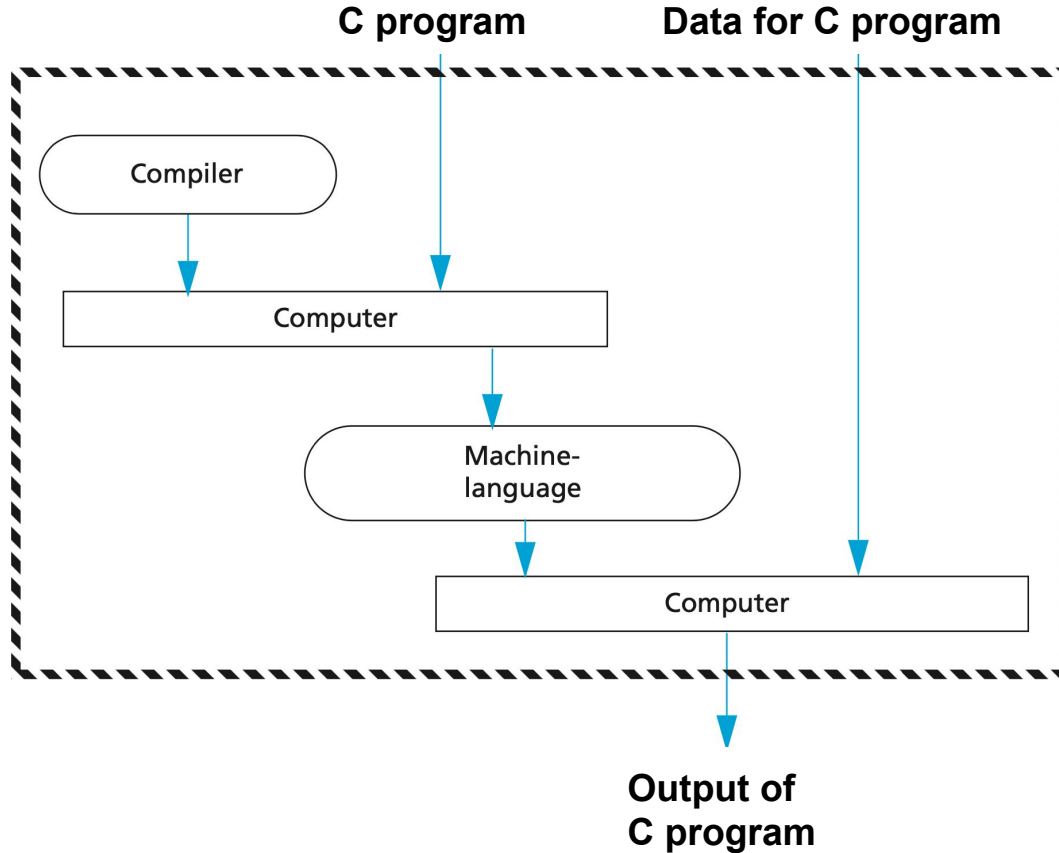


Compilers

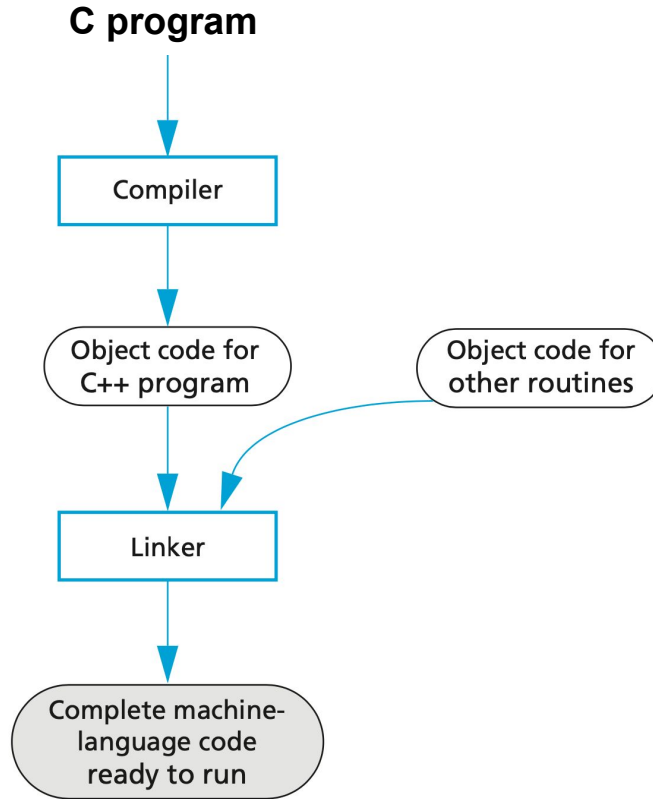
A program that translates a high-level language like C to a machine language is called a compiler.

- To avoid confusion, the input program is usually called the **source program** or **source code**, and the translated version produced by the compiler is called the **object program** or **object code**.

Compiling and running a C program (Basic)



Preparing a C program for Running (Linker)



Why C?

- Popular modern programming language
- Similar structure to many/most other popular languages (Java, C#, Perl, Python)
- Simple
- Fast
- Portability and Efficiency

Understand the Machine (Think in C)

Why C

Some programming history

- C developed as improvement on B
- B developed as improvement on ...
- BCPL – Basic Computer Programming Language
- Various languages before BCPL – ADA, COBOL, FORTRAN

Course Outline

- Programming basics, input/output, arithmetic
- Conditional statements
- Loops
- Modularity – functions
- Pointers
- Complex data – arrays, strings, and classes

Throughout the semester:

- Proper programming style

END

A simple c program:

<https://www.programiz.com/c-programming/online-compiler/>

Self assessment questions

1. What are the five main components of a computer?
2. What would be the data for a program to add two numbers?
3. What would be the data for a program that assigns letter grades to students in a class?
4. What is the difference between a machine-language program and a high-level language program?
5. What is the role of a compiler?
6. What is a source program? What is an object program?
7. What is an operating system?
8. What purpose does the operating system serve?

Self assessment questions

9. Name the operating system that runs on the computer you use to prepare programs for this course.

10. What is linking?

11. Find out whether linking is done automatically by the compiler you use for this course.

Self assessment questions

12. An algorithm is approximately the same thing as a recipe, but some kinds of steps that would be allowed in a recipe are not allowed in an algorithm. Which steps in the following recipe would be allowed in an algorithm?

Place 2 teaspoons of sugar in mixing bowl.

Add 1 egg to mixing bowl.

Add 1 cup of milk to mixing bowl.

Add 1 ounce of rum, if you are not driving.

Add vanilla extract to taste

Beat until smooth.

Pour into a pretty glass.

Sprinkle with nutmeg

13. What is the first step you should take when creating a program

14. The program design process can be divided into two main phases. What are they?

15. explain why the problem-solving phase should not be slighted.

