# Chapter 1: Introduction to Computers and Programming

AP Computer Science A Prep

September 30, 2019

## Introduction to Computers and Programming

## Important Terms

- Computer: programmable machine designed to follow instructions
- Program: instructions in computer memory that solve a problem
- Programmer: person who writes instructions (i.e. programs) to make the computer perform a task

With out a programmer, there are no programs. With out programs, a computer cannot do anything.

## More Terms

- Language: A set of grammar which is used to define a solution to a problem.
- Compiler: A tool which converts a solution into software, which a computer can understand.
- Operating System: Software which allows you, the user, to execute other software and interacts with hardware.

## Hardware

## Main Hardware Component Categories

1. Central Processing Unit (CPU)
2. Main Memory
3. Secondary Memory (also called Storage)
4. Input Devices
5. Output Devices

## Input Devices

Input devices provide information to a computer. What are some input devices?

## Input Devices

Input devices provide information to a computer. What are some input devices?

- Keyboard
- Mouse
- Microphone
- Web camera
- Camera
- Joystick/Gamepad
- Drawing tablet
- Scanner

## Output Devices

Output devices are used to represent the results of a process. What are some output devices?

## Output Devices

Output devices are used to represent the results of a process. What are some output devices?

- Monitor
- Printer
- Speaker
- 3D Printer
- VR Goggles

## Main Hardware Component Categories

1. Input devices feed into the CPU and Main Memory.
2. CPU and Main Memory will often work with Storage memory.
3. Results from CPU will feed into the output devices.

See Figure 1-2 in your textbook for an overview of how the components interact.

## Central Processing Unit - CPU

Comprised of:

- Control Unit
  - Retrieves and decodes program instructions Coordinates activities of all other parts of computer
- Arithmetic & Logic Unit
- Hardware optimized for high-speed numeric calculation
- Hardware designed for true/false, yes/no decisions

## Main Memory

- Main Memory is **volatile**. In this context, **volatile** means that once the memory chip no longer has power, the data resets.
- Main Memory is called **RAM (Random Access Memory)**.
- Main Memory is a collection of logical gates.
- Each logic gate is called a **bit**. Each bit can retain an off or on position.
  - **off is usually interpreted as 0 or false**
  - **on is usually interpreted as 1 or true**
  - A collection of 8 bits is called a **byte**.

## Memory Addressing

- Individual bits don't have addresses.
- Each **byte** is assigned a unique **address**.
- The first address on a chip is 0.
- The second address on a chip is 1.
- The third address on a chip is 2.
- The last address on a chip is $n - 1$, where $n$ is the number of bytes on the chip.

## Secondary Storage

- Secondary Storage is **non-volatile**: data is retained while the device is powered off.
- Examples of **non-volatile** memory:
  - Hard Drives
  - Optical: CD-ROM, DVD, Blu-ray
  - Flash Drives, Solid State Drives

## Categories of Software

- **System software**: programs that manage the computer hardware and the programs that run on them. Examples: operating systems, utility programs, software development tools
- **Application software**: programs that provide services to the user. Examples : word processing, games, programs to solve specific problems

## Programming Languages

## Programs

- A program is a set of instructions that the computer follows to perform a task.
- We start with an algorithm, which is a set of well-defined steps.

## Guiding the Green Circle

Imagine that you have two objects on your screen. Guide the Green Circle to the Blue Circle.

- Blue Circle
  - Is Blue
  - Cannot Move
- Green Circle
  - Is Green
  - Has a direction
  - Can Move Forward 1 Unit
  - Can Turn Left 90 degrees
  - Can ask for Distance to Blue Circle

## Machine Language

- Once we have an algorithm, we need to translate that algorithm into a programming language.
- After that, we ask the computer to translate that into a machine language.
- Computers only understand machine language.

## Machine Language

- Machine instructions are short sequences of binary code that perform a low-level operation in the computer.
- Rather than writing programs in machine language, most programmers use programming languages.

## Programming Languages

There are two types of languages.

- Low level language: machine language programming that differs from machine to machine.
- High level language: language allows for multiple low-level instructions to be combined into a single instruction. Much more human readable.

## Well-known programming languages

- C++
- C
- Java (You are here.)
- JavaScript
- Python
- Ruby
- Basic and Visual Basic
- Fortran
- C#

## Goals of Java

- Hava a syntax similar to C++. They did this to convert C++ programmers to Java.
- Be robust and secure.
  - Java doesn't have pointers, so there's no pointer syntax to understand.
  - Java manages memory for the programmer, so much of the security risks involving memory are eleminated.
  - Java memory management also handles memory leaks for the programmer.
- Compile once. Run anywhere.
- Be object-oriented.
- Be multi-threaded and concurrent.

## From a High-Level Program to an Executable File

1. Create file containing the program with a text editor or IDE.
2. Run the Java Compiler to convert source program into Java Bytecode. This produces a new file called the Java Class file and will have the extension ".class".
3. Run the Java Virtual Machine, which does two things:
    3.1 Converts the Java Bytecode into machine code. This is called **just-in-time compiling**.
    3.2 Executes the machine code.

The Java Virtual Machine takes care of the bytecode compiling and execution into what seems like a single step. It's doing multiple things.

## What is bytecode?

- Bytecode is a machine-level language for a theoretical computer which only exists as software.
- The "byte" in bytecode means that each instruction (not including arguments) is one byte long.
  - A byte is 8 bits.
  - $2^8$ is 256.
  - Java Bytecode is limited to 256 instructions.
- The Java Virtual Machine is a software application that will convert the bytecode into your computer's machine code on-the-fly and then execute that machine code.
- The purpose of bytecode is to allow Java programs to compile on any computer and then be executed on any other computer.
  - This is part of the success behind the video game Minecraft. Any computer with a JVM can execute Minecraft.

## IDE: Integrated Development Environment

- An integrated development environment, or IDE, combine all the tools needed to write, compile, and debug a program into a single software application.
- Examples are NetBeans, Microsoft Visual Studio, Eclipse, CodeWarrior, etc.
- We will be using NetBeans in this course.

## What are Programming Languages Made Of?

## What goes into a programming language?

- key words
- programmer-defined identifiers
- operators
- punctuation
- syntax

## Program HelloWorld

Code.

```
// This program prints "Hello, world!"
public class HelloWorld {
    public static void main(String[] args) {
        String message = "Hello, world!";
        System.out.println(message);
    }
}
```

## Key Words

- Also known as **reserved words**.
- Have an explicit meaning in Java which cannot be redefined.
- Cannot be used for any other purpose than that defined meaning.

## Keywords in the Hello World program

- public
- class
- static
- void

## Programmer-defined identifiers

- Names made up by the programmer used to identify parts of the program.
- Not part of the language.
- Used to represent variables, functions, classes, and more.

## Operators

- Used to perform operations on data.
- These are symbols and not words.
  - Arithmetic: + - * /
  - Assignment: =

## Other punctuation

- Other symbols used to organize the program.

## Syntax

- The rules of grammar that must be followed when writing a program.
- This is the combination of everything we've seen in the previous few slides.

## Variables

## Variables

- In math, a variable is an unknown quantity. Sometimes you must solve for the variable in an expression.
- In computer programming, a variable is usually something which is known while the program is running but unknown when the programmer is writing the code.
- Imagine that you've been tasked to write the code which posts a status update someone's social media profile. (Think of Facebook or Twitter.)
- You don't know what a user will say when they use your code.
- After the program is written and is running, the user can decide what they would like to post.
- The message of the status update should be a variable.
- The variable will allocate sufficient memory to hold the user's message.

## Variable Definitions

- In Java (this is not true for every language) you must write a variable definition (also called a variable declaration).
- The Java compiler will provide an error if it encounters a variable in an expression which has not been defined.
- There are many different sizes and types of variables and we will cover these in more detail in this course.

## Input and Output

## Input and Output

All programs from Microsoft Word to Facebook to Overwatch follow this general approach to processing data:

1. Get some data from an input device (such as the keyboard) or from a file.
2. Process that data using a combination of algorithm and secondary storage files.
3. Send the processed result to an output device (such as the monitor) or write the results to a file.

## The Programming Process

## The Programming Process (Old Way)

1. Clearly define what the program is to do.
2. Think a little.
3. Code a little.
4. Test. Return to Step 2 or stop if satisfied.

## The Programming Process (New Way)

1. Clearly define what the program is to do.
2. Test the code. Does it work? Stop if satisfied.
3. Refactor what you have written. If you haven't written anything, skip this step.
4. Think a little.
5. Code a little. Return to Step 2.

## Homework

## Your Homework Assignment

- Write three programs that produce ASCII art.
    - ASCII art is the creation of art using only text symbols.

## Your Homework Assignment