Week 1: Computer Science 1

How to speak machine

What is a computer?

A **computer** is a machine that can be programmed to carry out sequences of arithmetic or logical operations (computation) automatically. (Wikipedia)

What are the components of a computer?

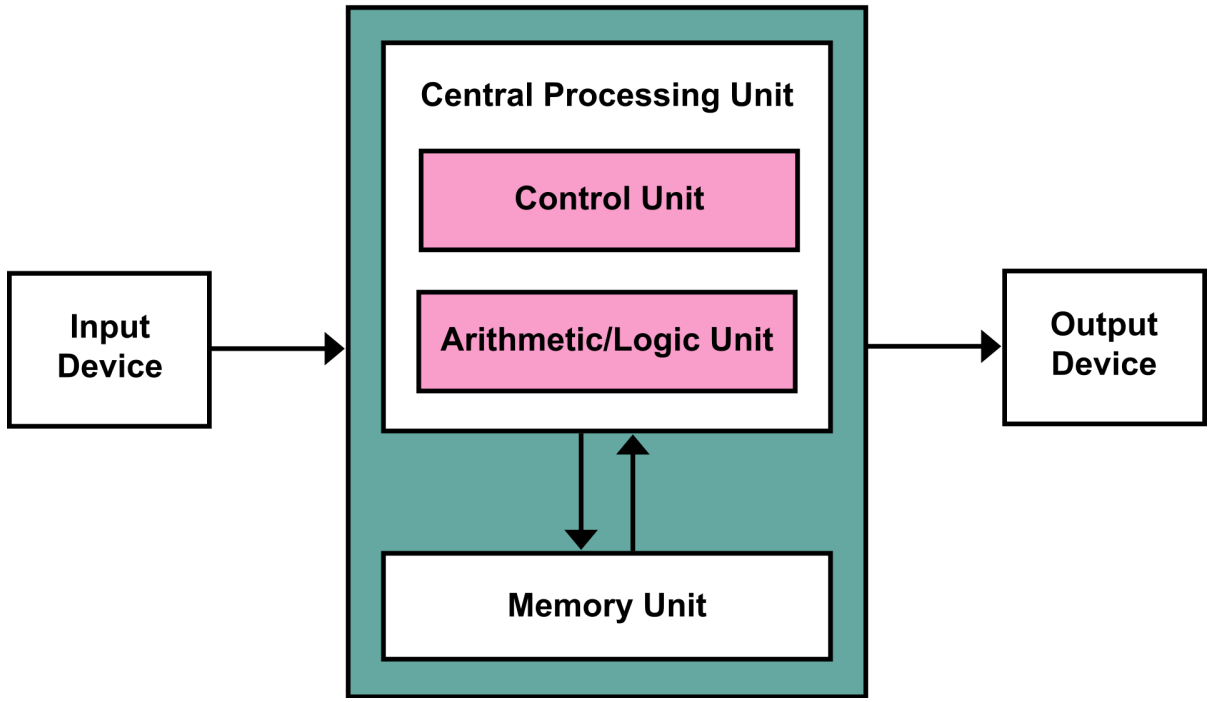
Computer Components

- Central Processing Unit (CPU)
  - Arithmetic Logic Unit (ALU): Add, Subtract, Multiply, Divide, Compare
  - Control Unit (CU): Fetch, Decode, Execute
- Memory
  - Random Access Memory (RAM). Main memory of the computer. Where the program and data are stored to be executed by the CPU. Volatile.
  - Hard disk. Secondary memory of the computer. Where the program and data are stored to be executed by the CPU. Non-volatile.
- Input/Output
  - Input: Keyboard, Mouse, Touchscreen, Microphone, Camera, etc.
  - Output: Monitor, Printer, Speakers, etc.

Von Neumann Architecture

1945 description for the EDVAC.

- A processing unit with both an arithmetic logic unit and processor registers
- A control unit that includes an instruction register and a program counter
- Memory that stores data and instructions
- External mass storage
- Input and output mechanisms



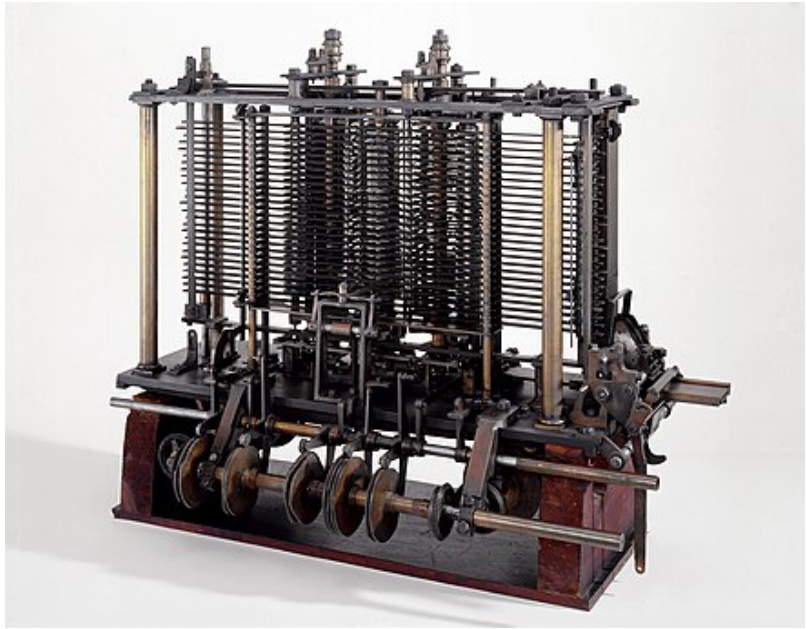
Jacquard Loom

The use of punched cards in the **Jacquard loom** (c. 1804) represented an early form of programmability. The pattern to be woven into the fabric was determined by the series of cards used, and different patterns could be produced by simply changing the card sequences. Thread was raised or lowered according to the hole positions along the card row, and the presence or absence of a hole indicated whether the thread was to be raised or lowered.

Analytical Engine

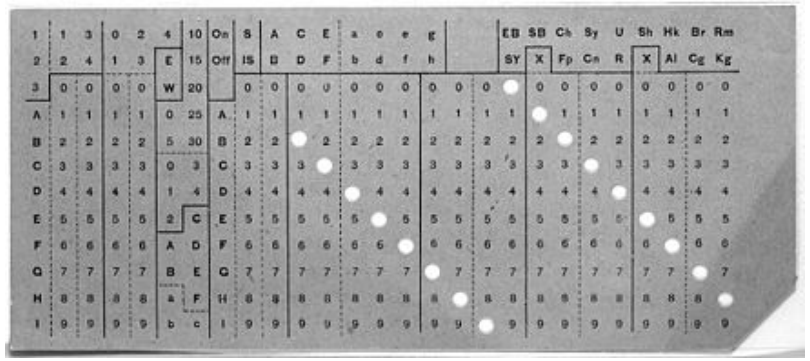
The **analytical engine** was a proposed mechanical general-purpose computer designed by English mathematician and computer pioneer Charles Babbage. It was first described in 1837 as the successor to Babbage's difference engine, a design for a simpler mechanical computer.

Lady Ada Lovelace, a mathematician and daughter of the poet Lord Byron, wrote the first computer program for the analytical engine.



Hollerith Punched Card

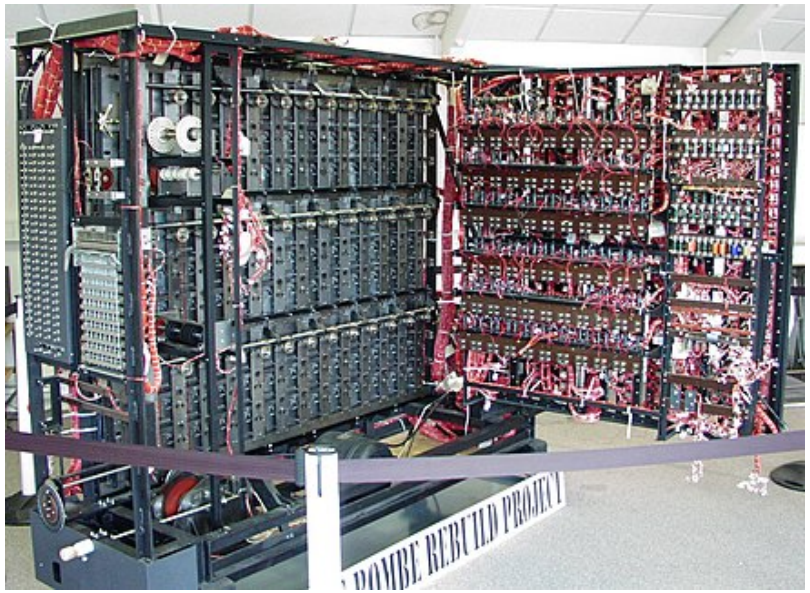
The **Hollerith card** was a type of punched card used for tabulating in the 1890 United States Census, invented by Herman Hollerith. It was the first such device that was used for this purpose and marked the beginning of the era of semiautomatic data processing systems. Later sold to IBM.



Bombe

The **Bombe** was an electromechanical device used by British cryptologists to help decipher German Enigma-machine-encrypted secret messages during World War II. **Alan Turing** was a major force in the development of the Bombe.

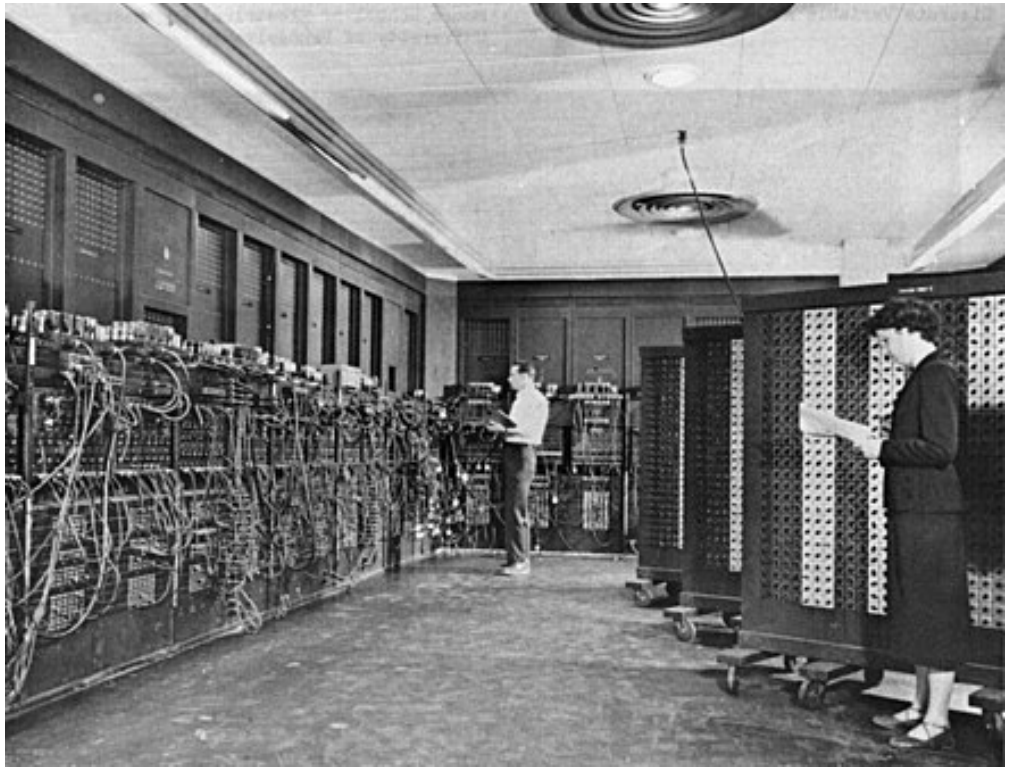
A Turing machine is a theoretical device that can simulate any algorithm or computation.



ENIAC

The **ENIAC** was the first electronic general-purpose computer. It was Turing-complete, digital, and capable of being reprogrammed to solve "a large class of numerical problems". It was the first programmable, electronic, digital computer.

Turing-complete refers to the capability of a system to perform any computation that can be done by a Turing machine.



What language does a computer speak?

Computers speak in **Binary** which is a **base 2** number system. It uses only **0 and 1** to represent numbers which is easy to represent with electrical circuits, 0 is off and 1 is on. All data and software in a computer is represented in binary.

Decimal: Base 10

Position	Weight
Ones	10^0
Tens	10^1
Hundreds	10^2
Thousands	10^3
Ten-thousands	10^4

235 = (2 * 10^2) + (3 * 10^1) + (5 * 10^0)

235 = 200 + 30 + 5

Binary: Base 2

Position	Weight
Ones	2^0
Twos	2^1
Fours	2^2
Eights	2^3
Sixteens	2^4

7 = (1 * 2^2) + (1 * 2^1) + (1 * 2^0)

7 = 4 + 2 + 1

Let's convert the decimal number 235 to binary.

First, what is the largest power of 2 that is less than 235?

Power	Value
2^0	1
2^1	2
2^2	4
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256
2^9	512
2^10	1024

The largest power of 2 that is less than 235 is 2^7 = 128.

Let's put a 1 in the 2^7 position:

10000000

Now we need to subtract 128 from 235 to get the remainder.

235 - 128 = 107

What is the largest power of 2 that is less than 107?

The largest power of 2 that is less than 107 is 2^6 = 64.

Let's put a 1 in the 2^6 position:

11000000

Now we need to subtract 64 from 107 to get the remainder.

107 - 64 = 43

What is the largest power of 2 that is less than 43?

The largest power of 2 that is less than 43 is 2^5 = 32.

Let's put a 1 in the 2^5 position:

11100000

Now we need to subtract 32 from 43 to get the remainder.

43 - 32 = 11

What is the largest power of 2 that is less than 11?

The largest power of 2 that is less than 11 is 2^3 = 8.

Let's put a 1 in the 2^3 position:

11101000

Now we need to subtract 8 from 11 to get the remainder.

11 - 8 = 3

What is the largest power of 2 that is less than 3?

The largest power of 2 that is less than 3 is 2^1 = 2.

Let's put a 1 in the 2^1 position:

11101010

Now we need to subtract 2 from 3 to get the remainder.

3 - 2 = 1

What is the largest power of 2 that is less than 1?

The largest power of 2 that is less than 1 is 2^0 = 1.

Let's put a 1 in the 2^0 position:

11101011

Now we need to subtract 1 from 1 to get the remainder.

1 - 1 = 0

We are done! The binary representation of 235 is 11101011.

Now you try converting the decimal number 411 to binary.

The binary representation of 411 is 110011011.

The CPU can only understand binary and processes data in this format. Bit's are the smallest unit of data that a computer can process.

A **bit** is a single binary digit, either 0 or 1.

A **byte** is a group of 8 bits.

A **word** is the number of bits that can be transferred from memory to the CPU in a single operation. This is usually 32 or 64 bits.

bit -> byte -> word

Storage
bit -> byte (8 bits) -> kilobyte (1024 bytes) -> megabyte (1024 kilobytes) -> gigabyte (1024 megabytes) -> terabyte (1024 gigabytes)

What is Programming?

Programming (coding) is the process of creating a set of instructions that tell a computer how to perform a task.

There are multiple programming languages that can be used to create these instructions. Such as Java, Python, JavaScript, C++, C, Assembly, etc.

In this course we are going to use the **Java language**, a **high-level** programming language.

Different types of programming languages:

- **Low-level** programming languages are closer to machine language. They are harder to read and write than high-level languages. Examples are Assembly and C.
- **High-level** programming languages are closer to human language. They are easier to read and write than low-level languages. Examples are Java, Python, C++, JavaScript, etc.

We are close to being able to write our first Java program. But first we need to create a file to store our program.

In VS Code go to File -> New File. Then create a new file called **HelloWorld.java**. We need to save this file with a name that ends in **.java**. This is the file extension for Java programs.

You should now see the **HelloWorld.java** file in the **Explorer** pane on the left side of the screen and the text editor should be open with the file loaded.

Write out first line of code!

When programming you have to follow a set of rules called **syntax**. If you don't follow the syntax rules the program will not run.

Our first Java program will print "Hello World!" to the screen. The most famous program in the world!

```
System.out.println("Hello World!");
```

There you go, you just wrote your first line of code!

Java is case sensitive, so **System.out.println** is not the same as **system.out.println**.

Notice the semicolon at the end of the line. This is required at the end of every statement in Java.

Java code needs to be formatted in a specific way. For example, you can insert any number of spaces between two **tokens** (words, numbers, symbols) and the program will still run.

```
System.out.println(20+3);
System.out.println(20 + 3);
System.out.println(20 +3);
System.out.println(20+ 3);
```

Each one of these will print 23 to the screen.

You may not insert spaces in the middle of a token.

```
System.ou t.pri ntln(2 0+3);
```

System, out, println, 20, and 3 are all separate tokens.

Structure of a Java Program

Let's combine what we have learned so far to write a program that prints "Hello World!" to the screen.

Java programs are made up of **classes**. A class is a template for creating objects and the basic unit in a Java Program. We will learn more about objects later in the course.

```
class class name
{
     .
     .
     .
}
```

Within the class we have **methods**. A method is a named sequence of Java statements.

```
method header
{
    statements
    .
    .
}
```

The **header** of a method contains the method name and any parameters that are passed to the method. We will learn more about parameters later in the course.

The **body** of a method contains the statements that are executed when the method is called.

Each line of code we have written so far is a statement. All statements must end with a semicolon.

The simplest Java program is a single class with a single method.

```
class clase name
{
    method header
    {
        statements
        .
        .
    }
}
```

Let's use this structure to write our first Java program.

```
class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}
```

Let's break down each part of this program. There will be a lot of new terms and concepts in this section. Don't worry if you don't understand everything right away. We will cover all of these concepts in more detail later in the course.

```
class HelloWorld
```

class: Indicates the beginning of a class.

HelloWord: This is the class name. The class name must match the file name. In this case the file name is HelloWorld.java. Notice that the class name starts with a capital letter. This is a convention that we will follow throughout the course.

```
{
    public static void main(String[] args)

}
```

{ }: Indicates the beginning and end of the class body.

public: This is the start of the header for the main method. Public means that the method can be accessed from outside the class and is unrestricted.

static: Indicates the we can excute the method without creating an object of the class.
void: Indicates that the method does not return a value.

main: This is the name of the method. The main method is the entry point for all Java programs. The main method is where the program starts executing. Notice that the method name starts with a lowercase letter. This is a convention that we will follow throughout the course.

( ): Indicates the beginning and end of the method header.

String[]: This is data type. The square brackets indicate that this is an array of Strings.

args: This is the name of the parameter.

```
{
    System.out.println("Hello World!");
}
```

{ }: Indicates the beginning and end of the method body.

System.out.println: This is a method call. It calls the println method of the out object of the System class.

("Hello World!"): This is the parameter that is passed to the println method. It is a String. We will

```
class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}
```

```
class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}
```

Notice the formatting of the code. The method header is indented one tab and the method body is indented two tabs. This allows us to easily see where the method begins and ends.

Having correct formatting is important for readability. It makes it easier for you and others to read and understand your code. Get used to formatting your code correctly now, it might take a little longer but it's a very important habit to develop.

Compile and run your first Java Program

When your computer runs a Java program it first needs to convert the Java code into a format that the computer can understand. This is called **compiling** the code. The Java compiler is called **javac**.

It converts our human-readable code into **bytecode**. Bytecode is a set of instructions that can be executed by the Java Virtual Machine (JVM). The JVM is a program that runs on your computer and executes Java programs.

The JVM is different for each operating system. This means that the same Java program can run on Windows, Mac, and Linux.

Let's compile our first Java program. Open a terminal window and navigate to the folder where you saved the **HelloWorld.java** file.

Your should already have a terminal opened in VS Code but if you don't you can open one by going to Terminal -> New Terminal. You should see a terminal window at the bottom of the screen and it's opened to the **hello-world** folder. If not, use the command line to navigate to the **hello-world** folder.

Now enter the following command to compile the program.

```
javac HelloWorld.java
```

You should see a new file in the **hello-world** folder called **HelloWorld.class**. This is the compiled bytecode version of our program.

Now we need to run the program. Enter the following command to run the program.

```
java HelloWorld
```

You should see the following output in the terminal.

```
Hello World!
```

Congratulations! You just wrote and ran your first Java program!