

CECS 174 – Lecture 1 – Introduction

What is a Computer? –

Computers are made up of two key elements:

1. Hardware - the physical components of the computer, such as the monitor, keyboard, CPU, hard drive, memory, etc.
2. Software - the instructions that make it work, also known as programs.

What is a Program? –

A program is a set of instructions processed by the CPU that tells the computer exactly what to do, one step at a time, in the stated order. Most programs usually involve receiving input, processing data, and then outputting the results.

Programs are written to solve a problem or to perform a task. In order to do this, a problem is broken down into smaller and smaller steps until just individual instructions that the computer can understand are left.

How Do Programmers Tell a Computer What to do? –

First off, you should know that computers aren't really very smart – they are actually only capable of doing four basic tasks. They can:

1. Hold data – stores the data in memory.
2. Move data – moves data from one memory cell to another.
3. Add data – adds a value to another value.
4. Compare data – compares two values to determine their equivalence.

Second, computers only understand binary code (a low-level language where instructions are made up of a set of 1's and 0's). Binary can be extremely difficult for programmers to write in, and is even more difficult to read.

Fortunately for us, high-level programming languages, such as Java, were created to reduce the difficulty of having to work directly with the computer. High-level languages come with a predefined syntax and a set of commands for the programmer to use, which give the programmer a way to write code that is closer to how a human thinks. This allows anyone to write a program as long as they are able to break down the task into instructions that the computer can understand.

Programming Languages allow you to process data by using the structural programming paradigm (Dijkstra):

1. Sequential – each step follows the last.
2. Branch – a comparison or decision determines if code is executed.
3. Loop – instructions are repeated until a test condition is reached.
4. Module – steps are encapsulated so they can be easily re-used.

There is also the object oriented programming paradigm, which consists of breaking up a problem into different objects, each of which has its own set of data, tasks, and functions.

Once a programmer finishes writing their code, it is then compiled into binary instructions. When the program is run, the computer executes these instructions.

How is a Program Written? –

A program is written by going through several phases, some of which may need to be repeated or revisited throughout the process:

1. Understand the Problem – if the program is already defined, go through and read the requirements until you fully understand what the problem is asking. If there are any parts that you don't understand, you may have to research or examine them until you do. If it is not defined, then you must figure out and define what the program will do.
2. Break Down the Problem – examine the problem to see if there are any smaller tasks that can be broken apart and worked on separately. It is useful to find as many as you can early on since they can be written to be reused, which means that you will have less work to do.
3. Generalize – try to write your programs in general terms. For example, don't write a program or a module that only prints out the answer to $3 + 5$, write it to display the result of adding any two numbers.
4. Optimize – simplify tasks, modularize, make your code more efficient. Think about your code in terms of size, space, and run time:
 - a. Size – your code takes up memory, the less code there is, the less space it takes up. Eliminate unnecessary instructions when possible.
 - b. Space – reserving memory for values that your program uses also takes up memory space, the fewer you use, the better.
 - c. Run Time – processing the instructions your program takes up time. Figure out how to make your program run in the least amount of time.
5. Readability – use good variable names and add spacing and comments wherever necessary to make your code more readable by other people.

What is the Programming Process? –

There are several steps a programmer goes through to develop a program:

1. Requirements – create or read the requirements of the task.
2. Planning and Design – break the task up into pieces, create algorithms.
3. Writing the Code – choose a language, and follow its syntax.
4. Compile the Code – translate the code so the computer can understand it.
5. Testing and Verification – run the program to debug and check for errors.
6. Place into Production – distribute and sell the program.
7. Maintenance – update the program as necessary.

Who Can Program? –

Anyone with an innate curiosity of how problems are solved combined with logical way of thinking. Problems can exist in any field; in order to be able to solve these problems you must be willing and able to learn almost anything.

What is Java? –

Java is a high-level language similar to C++ that can be used to write almost any type of program for a computer. Before Java, a program had to be compiled specifically for a particular operating system. Now each operating system can have a Java Virtual Machine (JVM), which a Java program is run on. This means that any operating system with a JVM can execute any Java program.