



# Programming Fundamentals

## Introduction to Computer Programming

### 3<sup>rd</sup> Lecture

#### 1. Introduction

In another subject in this class, you have already learned an important lesson of computing: Software (programs) rules the hardware (the physical machine). It is the software that determines what any computer can do. Without software, *computers would just be expensive paperweights*. The process of creating software is called **programming**, and that is the main focus of this lecture. Computer programming is a challenging activity. Good programming requires an *ability to see the big picture while paying attention to minute detail*. Not everyone has the talent to become a first-class programmer, just as not everyone has the skills to be a professional athlete. However, virtually anyone can *learn how to program computers*. With some patience and effort on your part, this course will help you to become a programmer. There are lots of good reasons to learn programming. Programming is a fundamental part of computer science and is, therefore, important to anyone interested in becoming a computer professional. But others can also benefit from the experience.

#### 2. Computer Programming

Today, most people don't need to know how a computer works. Most people can simply turn on a computer or a mobile phone and point at some little graphical object on the display, click a button or swipe a finger or two, and the computer does something. An example would be to get weather information from the net and display it. How to interact with a computer program is all the average person needs to know. But, since you are going to learn how to write computer programs, you **need to know a little bit about how a computer works?** Your job will be to instruct the computer to do things.



**process** / Noun:

A series of actions or steps taken to achieve an end.

**procedure** / Noun:

A series of actions conducted in a certain order.

**algorithm** / Noun:

An ordered set of steps to solve a problem.

Basically, writing programs (computer software or applications) involves describing processes, procedures; **it involves the authoring of algorithms.** Computer programming involves developing lists of instructions - the **source code representation of program**. The stuff that these instructions manipulate are different types of objects, e.g., numbers, words, images, sounds, etc... Creating a computer program can be like composing music, like designing a house, like creating lots of stuff.

A fair question you may have is "**Why should I learn how to program a computer?**"

*An important reason to consider learning is that the concepts underlying this will be valuable to you*, regardless of whether or not you go on to make a career out of it. One thing that you will learn quickly is that a computer is very dumb, but obedient. It does exactly what you tell it to do, which is not necessarily what you wanted. Programming will help you learn the importance of clarity of expression.

"A deep understanding of programming, in particular, the notions of successive decomposition as a mode of analysis and debugging of trial solutions results in significant educational benefits in many domains of discourse, including those unrelated to computers and information technology per se."

(Seymour Papert, in "Mindstorms").

"It has often been said that a person does not really understand something until he teaches it to someone else. Actually, a person does not really understand something until after teaching it to a computer, i.e., express it as an algorithm."

(Donald Knuth, in "American Mathematical Monthly").

Therefore, we can define a **Computer programming** as the process of writing instructions that get executed by computers. The instructions, also known as code, are



written in a programming language which the computer can understand and use to perform a task or solve a problem.

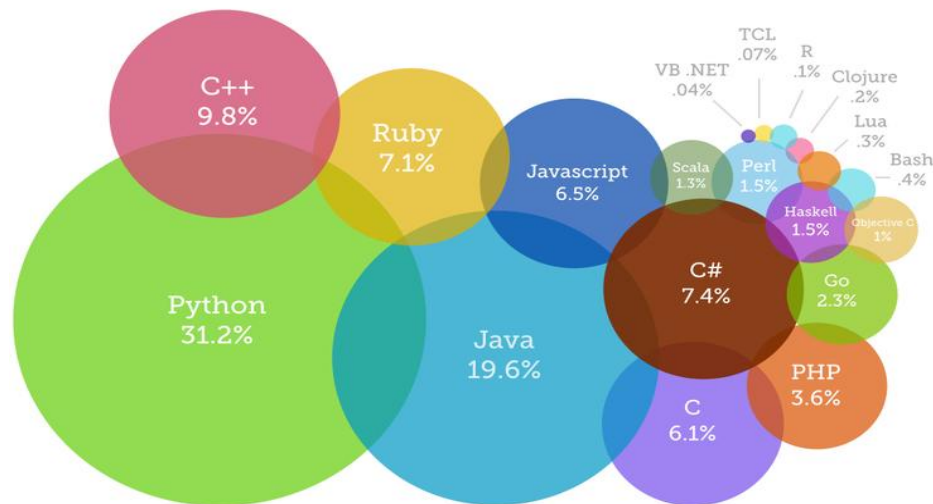
So... just what do instructions a computer understands look like? And, what kinds of objects do the instructions manipulate? By the end of this course you will be able to answer these questions. But first let's try to write a program in the English language. Remember that “Writing software, computer programs, is a lot like writing down the steps it takes to do something”.

### 3. Programming Language

A **programming language** is a vocabulary and a set of grammatical rules for instructing a computer or computing device to perform specific tasks. The term programming language usually refers to high-level languages, such as Python, C, C++, R, Java, C#, Swift, etc.

*Each programming language has a unique set of keywords* (words that it understands) and a *special syntax* for *organizing program instructions*.

According to IEEE Spectrum's interactive ranking, Python is the top programming language of 2017, followed by C, Java, and C++. Of course, the choice of **which language to use** depends on the type of computer the program is to run on, what sort of program it is, and the expertise of the programmer.



Most Popular Programming Language of 2015



Language Rank	Types	Spectrum Ranking
1. Python		100.0
2. C		100.0
3. Java		99.4
4. C++		96.9
5. C#		88.6
6. R		88.1
7. JavaScript		85.3
8. Go		75.7
9. Swift		74.3
10. Ruby		72.0

Most Popular Programming Language of 2017

We have mentioned here 4 excellent programming languages of 2017 which you should learn and have a better idea.

- 1- **Python:** For becoming skilled at all-in-one language, you should begin learning Python language that has the ability to expand *web apps, data analysis, user interfaces*, and much more, and frameworks are also available for these tasks. Python is utilized by bigger companies mostly that can evaluate vast data sets, thus this is a huge chance to learn it and be a Python programmer.
- 2- **Java:** Java is considered as the *perfect language for the developers and programmers to learn*. It's one of the most widely adopted programming languages, used by some 9 million developers and running on 7 billion devices worldwide. It's also the programming language *used to develop all native Android apps*. Java's popularity with developers is due to the fact that the language is *grounded in readability and simplicity*. Java has staying power since it has long-term compatibility, which makes sure older applications continue to work now into the future.



- 3- **C++:** It is a *highly efficient and flexible language*, first created back in 1985. It has remained in high demand due to its *performance, reliability, and variety of contexts* you can use it in. Plenty of large systems have been created and maintained successfully using C++, including the likes of Microsoft, Oracle, PayPal, and Adobe. A career with C++ typically involves developing desktop applications, especially for performance-intensive tasks.
- 4- **C#:** C# (pronounced C sharp) is a *general-purpose, multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines*. It was developed around 2000 by Microsoft for a wide range of enterprise applications that run on the .NET Framework. C# is one of the programming languages designed for the Common Language Infrastructure.

Someone who can write computer programs or in other words, someone who can do computer programming is called a Computer Programmer.

**<Best Regards>**

*Dr. Raaid Alubady*