**English homework III**

**Describing something we work with in our profession**

As a software developer, one of the essential tools you need in order to work is, a programming language, or moreso the knowledge of one. Even though a beginner, one of the more intuitive and advanced programming languages I enjoy using is Python.

Python is an interpreted, high-level, general-purpose programming language. It’s a multi-paradigm programming language, which basically means that it will suit the needs of many programmers. Object-oriented and structured programming are fully supported, and many of its features support functional and aspect-oriented programming including metaprogramming and metaobjects.

This programming language uses dynamic typing and a combination of reference counting, and a cycle-detecting garbage collector for memory management, which many people who come from C like languages will love, because who likes to take care of pointers?

People who use Lisp are also going to be very familiar with Python. It has filter, map and reduce functions, list comprehension, dictionaries, sets and generator expressions. The standard library has two modules: itertools and functools. Basically, you won’t be a stranger to it.

A very unique idea developers of Python had, and also implemented, was not to build all its functionalities into its core, but to have it highly extensible. This means you won’t need much memory space or resources to run the very core of Python, and you can pick and choose which programmable interfaces you will add to it. It’s a very small core language with a large standard library and an easily extensible interpreter.

Simple, customizable, intuitive, with less-cluttered syntax and grammar; Python has it all, so there’s no reason why you should not try it for yourself, and maybe make it your main programming language.

For all of you newcomers, or even for people whose profession has nothing to do with programming, well… I guess math is the only requirement... you can utilize Python as well. You see, a simple and intuitive syntax is pretty easy to learn and understand, and the customization and a wide range of libraries you can add can turn this language from a developer’s world into a researcher’s world within a few clicks. Python is massively used in the world of scientific computing, linear algebra, combinatorics, numerical mathematics, number theory, calculus, chemistry etc. Basically, if you need an advanced calculator, Python will be your best buddy.

Describing something you can’t touch, to general public with no knowledge on the topic is very difficult, especially because you can’t get too professional and detailed, cause very few would understand you, and those who do, would probably get bored very easily, but for those brave enough to get into the specifics… These are Python’s syntax and semantics:

Again, meant to be very readable, its formatting is visually uncluttered, and it often uses English keywords, where other languages use punctuation. Unlike many other languages, it does not use curly brackets as block separators, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than Pascal or C like languages.

Python uses whitespace indentation to determine blocks. An increase in indentation comes after certain statements, a decrease in indentation signifies the end of the current block of code. Thus, the program’s visual structure accurately represents the program’s semantic structure. This feature is sometimes termed the off-side rule, which some other languages share, but most languages indentation doesn’t have any semantic meaning.

Python’s statements include:

- The *assignment* statement (token “=”, the equals sign),

- The *if* statement, which conditionally executes a block of code, along with else and elif (contraction of else-if),

- The *for, while-do, do-while loops* used to run a block of code multiple times,

- The *try* statement, which allows exceptions raised in its attached code block to be caught and handled by *except* clauses, as well as *finally* block, that always runs regardless of how the block exits,

- The *raise* statement, used to raise a specified exception or re-raise a caught exception,

- The *class* statement, which executes a block of code and attaches its local namespace to a class, for use in object-oriented programming,

- The *def* statement, which defines a function or method,

- The *break*, continue and pass statements, used to break/skip/create an empty, block of code,

- The *assert* statement, used during debugging to check for conditions that ought to apply.

- The *yield* statement, which returns a value from a generator function

- The *return* statement, used to, you guessed it, return a value from a function,

- The *import* statement, which is used to import modules whose functions or variables can be used in the current program, etc.

I strongly encourage you, if you are interested in other details about syntax, to visit the official [Python’s website](https://www.python.org/) and read all about it.

In retrospect, programming language Python is something every computer whiz should at least try, it’s very versatile, easy to use, and above all, a very powerful tool in the world of programming, if utilized properly.