#1 Assignment - CMPT 383

Luiz Fernando Peres de Oliveira - 301288301 - lperesde@sfu.ca

September 12th, 2017

1. What is functional programming? What is procedural programming?

Functional programming is a very famous programming paradigm (emphasized in academia rather than in commercial software development) that has its structure based on expressions and/or declarations instead of statements, which are very common in another programming paradigm so called imperative programming. Functional programming evaluates computation as mathematical functions and discourages mutable data and changing the program state in order to reduce or eliminate side effects. Functional programming has its roots based on Lambda Calculus, which is a formal mathematical model for high order functions as atoms, meaning that the every single abstraction can be translated into a function that takes a function as its parameter and returns another function as its output (output here stands for a function's output and is not related to IO operations that are side effects). Functional programming languages: LISP, Scheme, Haskell and others.

On the hand, procedural programming bases itself in procedure, functions or sub-routines calls (not to be confused with mathematical functions related to functional programming). It also follows the imperative paradigm as it makes explicit reference to changing-state, performs a series of well-structured sequential steps and may have side effects. Systematically, it embraces a succession of statements and functions to perform a computational task or program. Pascal, C and Go are example of procedural languages.

2. What are packages? What are package managers? What are APIs? What are wrappers?

Software Packages: Packages are collections of files (e.g binary files, scripts and so on), that are bundled together and can be installed and removed as a group, Microsoft Office for example. Sometimes are also called libraries when they are related to code reuse. They may or not be dependent on other packages in order to function.

Package Managers: A package manager, as its name suggests, is a software used to install, uninstall and manage software packages. They make the task software code/binary distribution easier, designed to eliminate the need for manual installs and updates. Homebrew is an example of package manager.

API: Application Programming Interface is a set of functions and protocols for software applications by providing specifications that software programs can follow to communicate with each other. Examples on that would be: REST, OpenGL and others.

Wrappers: Wrapper is a software that functions as an adapter for other softwares. An example would be software for a bank that would encapsulate a cryptography package/library in its source code.

3. How do "programming languages" differ from other languages? What other types of languages are there?

Programming languages are designed to be easily understood by machines and people, having the purpose of interfacing this interaction. They are very simple if compared with natural languages, such as English and Spanish. Also, programming languages are usually written, unambiguous and have a well-specified grammar, whereas natural languages may be spoken and be related to culture and region, being very ambiguous. There are various types of languages such as programming, natural, sign and others.

4. For those that code in an imperative language, why don't they all use Java? Why do multiple imperative languages exist (e.g., C, C++, Java, Objective-C, Python, etc.)?

Most programming languages try to solve specific problems rather than general ones. That is to say, comparing programming languages is a rather hard task. One would say that C is better than Java because it is faster and more optimized and other would say that Java is better than C because it is easier to handle memory or to develop commercial software. Many programming languages exist because, as any other kind of language, they evolve and are improved with time. With better processors, we are now able to have better abstractions in our programming environment (remembering the the more abstractions the farther from the bare metal we are and therefore, longer we take to solve a given problem). Not all people program in Java because people have different needs and they are trying to, many times, solve different problems.

5. When is runtime type checking advantageous?

Type checking can be handy when you want to eliminate certain types of errors before running a program.

Imagine that you have a function f, that takes an parameters of type **int** and returns an **int**. Type-checking will guarantee that f is only operating on **ints**. So before you run the program, you will catch any case where you call function f on the wrong type, which will cause a compiler error.

6. What are the least popular and most popular programming languages used in industry? Explain why their popularity is as such.

The most prevalent programming languages on the industry are **imperative languages**, such as Javascript, Python, PHP, Java and C#. They are popular because they have in general very productive, have many frameworks and have a big community on sites as Stack Overflow, helping beginners in problem solving.

In contrast, the least popular languages in the industry are esoteric programming languages, such as Brainf**k a type o programming languages that is seen as a "joke".

7. What is the average annual pay for computer scientists that use functional languages? What is the average annual pay for computer scientists that use imperative languages? Which language has the highest annual salary for computer scientists?
The average annual pay for computer scientists that use functional languages is $\$68,323$, the average annual pay for computer scientists that use imperative languages is $\$69,108$. Java has the highest annual salary. According to ChallengeRocket.com is goes over $\$130,000$.

8. What is the most advanced or cutting-edge programming language? What makes it so?
We know that programming languages like C#, Java and C++ will continue to be the backbone of any development in IT, but there have popped up some interesting programming languages that have achieved many good reviews and high ratings such as Go, Swift, Hack, Julia, Scala and others. These languages are ranked as the cutting-edge programming languages because they are developed by the best compiler developers in the world with the latest technologies. Go and Swift for example are supported by Google and Apple respectively and are growing substantially in the market as more and more people learn them.

9. What is the relationship between mathematics and computer science?

Mathematics is one of the foundations of computer science. As computer science is a subfield of mathematics, all computer science applications require mathematics to some degree. If you take the field of logic, for example, you may see yourself completely submerged by discrete mathematics and if you make a game, you will need a very good understanding of linear algebra and so on.

10. What are Domain Specific Languages?
A domain-specific language (DSL) are languages with very specific goals in design and implementation. Unlike a general-purpose language such as UML, a domain-specific language is designed to solve problems in a particular domain or universe. Each DSL is much better than a general-purpose language for describing operations on its own domain, but much worse for describing ideas that are outside its own scope.

11. What is the fastest programming language? What is the best programming language?

It is very hard to compare languages and have precise benchmarks regarding the **fastest** and the **best** programming languages. That happens because such benchmarks cannot compare or cannot create general topics in which one could use as a mean of measurement. However, the fastest languages are always the closest to the "metal". Thus, Assembly would be the fastest (not considering how hard it is to complete a task). Likewise, the best language varies from person to person, considering different backgrounds.

12. What are side-effects?

Side effects are a very likely to happen when a program interacts with the external structures and/or global variables, e.g on IO procedures, file handling, networking and others. Side effects are nothing but simply the modification of some kind of state. Side effects are often related to the programming paradigms. The imperative programming paradigm, for example, is known for its frequent utilization, and as said on question #1, functional languages avoid to use them, as it can lead to many problems (bugs and such).

13. What is syntax? Semantics? Semiotics?

Syntax: refers to the spelling and grammar of a programming language.
Semantics: refers to the meaning of a programming language.
Semiotics: area of study which is fundamentally concerned with how meanings are expressed and interpreted.

14. What is concurrent programming? What is event-driven programming?

Concurrent Programming: can be outlined as many (at least two) processes running throughout the same period of time. Differently from parallel programming, the execution does not need to happen at the same instant, thus, it consists of processes that overlap their lifetime. The main objective of concurrent programming is to model processes that happen concurrently. Many clients fetching data in the same time span from a server (not necessarily in parallel) would be a good example of how that works.
Event-driven Programming: when a computer program control flow is determined by event listeners, the occurrence of these events are observed (or monitored) and executed by an, so called, event handler, which is typically a callback (delegate function) or method that determines the course of the program through user actions such as clicks or key presses or through messages sent by other threads and/or programs.

15. What is the difference between algebra and calculus?

Algebra is a branch of pure mathematics, which deals with operations and relations of mathematics, and their respective rules. It focuses on different rules, and what is the outcome when operations are derived from other things, except numbers.

Calculus is another branch of mathematics, which deals with the function, limits, derivative and integrals and infinite series.

Calculus is different subject to understand, where as Algebra is much simpler. These two are different from each other as calculus is the study of change and Algebra deals with the relations. One (calculus) is concerned with relations and other measures the rate of change, the other (algebra) is the language of math. It deals with the rules, constructs, and nuances of the language of math.

REFERENCES

[1] http://en.wikipedia.org/wiki/Functional_programming

[2] https://stackoverflow.com/questions/23277/what-is-the-difference-between-procedural-programming-and-functional-programming

[3] https://en.wikipedia.org/wiki/Procedural_programming

[4] http://searchmobilecomputing.techtarget.com/definition/package-manager-or-package-management-system-PMS

[5] https://aptitude.alioth.debian.org/doc/en/pr01s02.html

[6] https://stackoverflow.com/questions/7440379/what-exactly-is-the-meaning-of-an-api

[7] https://en.wikipedia.org/wiki/Wrapper_function

[8] https://linguistics.stackexchange.com/questions/8612/any-difference-between-natural-and-programming-languages

[9] https://cs.stackexchange.com/questions/451/why-are-there-so-many-programming-languages

[10] https://www.quora.com/Why-is-type-checking-important-in-programming-languages-and-how-should-one-choose-between-dynamically-and-statically-typed-languages

[11] https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages

[12] https://msdn.microsoft.com/en-us/library/bb126278.aspx

[13] https://softwareengineering.stackexchange.com/questions/40297/what-is-a-side-effect

[14] https://en.wikipedia.org/wiki/Side_effect_%28computer_science%29

[15] http://www.cambridge.org/ca/academic/subjects/psychology/cognition/semiotics-programming?format=PB&isbn=9780521736275

[16] https://en.wikipedia.org/wiki/Concurrent_computing

[17] https://www.reddit.com/r/explainlikeimfive/comments/27zsra/eli5whats_the_difference_between_algebra_and/