

Programming Languages: Principles and Paradigms

Chapter 1

Exercises 1-4, 10

1.1

(a) Eiffel is object oriented and has static/strong typing. The language has influenced Java, C#, Ruby, and others. Readability is important. There is an “Eiffel programming method,” based on a set of programming principles. It was influenced by Ada, Simula, and Z and was released in 1986. Eiffel discourages procedural code and encourages declarative statements. Programs are collections of classes, called 'clusters.'

(b) Perl supports several programming paradigms – procedural, OO, functional – and uses an interpreter. It emphasizes practicality over elegance, and was originally developed for manipulating text. It derives broadly from C, and has in turn influenced Python, PHP, Ruby, and others. Perl's main goal is to make “efficient use of expensive programmers” rather than “efficient use of expensive hardware.” Perl uses “subroutines” rather than “functions.” It was released in 1987.

(c) Python emphasizes very clear syntax and can be imperative or object-oriented (as well as other paradigms, to a lesser extent). It uses whitespace as delimiters. It is dynamically typed. It favors sparse, uncluttered grammar. It was released in 1991.

1.2

Unreadable C:

```
int main() {
    int x = 4, y = 6, z; (y < 10) ? ((x > 4) ? ((x == y) ? z = 5 : z = 1) : z = 99) : z = 2; return 0;
}
```

More readable:

```
int main() {
    int x = 4;
    int y = 6;
    int z;
    if (y < 10 && x > 4 && x==y)
        z = 5;
    else
        z = 1;
    return 0;
}
```

1.3

(a) Algol introduced code blocks using 'begin' and 'end' and nested functions. Cobol appears to have very convoluted control flow and no ability to user-define functions or types. Basic appears to be very procedural and limited.

(b) Flon is simply putting the responsibility of writing good code on the programmer rather than the tool.

1.4

In C, records can be returned from functions, but arrays cannot.

1.10

	Haskell	Python
Simplicity/readability	Haskell is generally considered a difficult language, but for those who are proficient with it, it can be very readable, but it can also be easily obfuscated. It is very syntactically concise.	Python emphasizes readability and simple grammar. It is widely known as a good first language to learn.
Clarity about binding	Haskell is strongly typed, and each function has a clear type path.	Python is dynamically typed, so certain data types can be ambiguous.
Reliability	Haskell is generally considered very reliable.	Python is also considered very reliable.
Support	Haskell has a large wiki with hundreds of examples, along with many third party tutorials	Python has extensive documentation as well as countless third party tutorials.
Abstraction	Haskell is functional. It is an accurate implementation of lambda calculus	Python is multi-paradigm; it can be OO, imperative, or even functional.
Orthogonality	Haskell is extremely orthogonal. It comes with type variables, type classes, and functions can be used as values	Python is also very orthogonal. Arithmetic operators have meaning for all data types.
Efficient implementation	The GHCi interpreter has extensive optimization capabilities.	Python has many extensions such as 'itertools' that increase efficiency, but generally put simplicity and readability over optimization.