Chris Fenton CNC - PLD Assignment 1

1.

a. Eiffel is a compiled, statically typed, object-oriented programming language. Some distinguishing features of Eiffel are garbage collection, a declarative style, and the design by contract paradigm. Eiffel, like Smalltalk, is a pure object-oriented language. Eiffel was influenced by Simula and Ada and has influenced popular languages like Ruby, Java, and C#.

b. Perl is an interpreted, dynamically typed, multi-paradigm programming language. Perl has been popular as a scripting language and its text processing features have been influential on other languages.

c. Python is an interpreted, dynamically typed, multi-paradigm programming language. Python is popular as both a scripting language and a general purpose programming language. Some key features of Python are whitespace for indentation and a general focus on readable code.

2.

An example of unreadable Java code:

```
class A {
  private int a = 0;
  private String b;

  public A(int b, String c) {
    this.a = b;
    this.b = c;
  }

  public int getA() {
    return this.a;
  }

  public String getB() {
    return this.b;
  }

  public static void main(String[] args) {
    A person = new A(21, "Steve");
    System.out.println("Name: " + person.getB());
    System.out.println("Age: " + person.getA());
  }
}
```

An example of more readable code:

```
class Person {
  private int age = 0;

  private String name;

  public Person(int age, String name) {
    this.age = age;
    this.name = name;
  }

  public int getAge() {
    return age;
  }

  public String getName() {
    return name;
  }

  public static void main(String[] args) {
    Person person = new Person(21, "Steve");
    System.out.println("Name: " + person.getName());
    System.out.println("Age: " + person.getAge());
  }
}
```

3.

a. Algol seems to provide a more structured syntax.

b. Flon’s statement speaks to the infalliable nature of humans. Even if there existed a perfect programming langugage, it’s users–humans–are imperfect.

4. Java’s class constructor is orthogonal–there is only one way to declare a class constructor (although you can just inherit the constructor from the super class). For loops are an example of non-orthogonality–there are multiple ways to create a loop.

10. Java vs Ruby

- Simplicity and readability:

  Ruby is more readable:

  ```
  def hello
    puts "Hello, World"
  end
  ```

  vs

  ```
  public void hello() {
    System.out.println("Hello, World!");
  }
  ```

- Clarity about binding

  Because Java is statically typed and Ruby is dynamically typed, Java is more clear about binding.

```
double x = 2.0;
return x * 2;
```

vs

```
x = 2.0
return x * 2;
```

- Reliability

  Because Java is statically typed, a large class of errors will be caught during compilation. Ruby is more prone to unexpected behavior as a result of it’s dynamic typing and relies more heavily on testing to catch errors.

- Support

  Both Java and Ruby are heavily documented and supported.

- Abstraction

  Both Java and Ruby are object oriented a provide a high level of abstraction. Both languages have great library support, but I find Ruby’s gem package manager to be prolific and easy to use.

- Orthogonality

  Java is more orthogonal that Ruby. Ruby is notorious for providing multiple ways to do something.

- Efficient implementation

  Java provides a more effiencient implementation. Java’s JIT compilation is more efficient that Ruby’s interpreter.