1. Let’s suppose you want to write a Java method that operates on lists; naturally, you want to write your method so that it is as reusable as possible. So you consider two options for the type of the method:

```java
void f(List<Object>) { … }
void f(List<?>) { … }
```

As is often the case, it is rare that one option is always better than another option. Explain the relative strengths and weaknesses of the two types above; provide code examples that demonstrate things you can do with one but not with the other.

2. Let’s suppose you are writing a system and can pick from one of two languages. Your comparison of these languages has boiled down to one issue: one language uses single-dispatching and operator overloading (like Java) while the other uses multi-dispatching but no operator overloading (like Cecil). Discuss the relative strengths and weaknesses of the two languages giving concrete examples to illustrate your points.

3. Give a code example where a covariant result type is useful. Give a code example where a covariant argument type is useful.