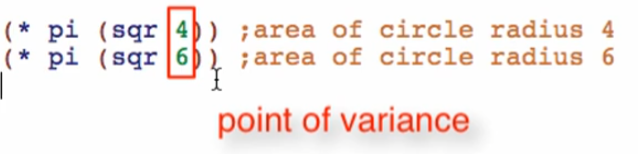
The simplest way to design abstract functions is to start with two or more examples of highly repetitive code. The first step in this way of designing an abstract function is to produce the working function definition.

1. Identify two highly repetitive expressions

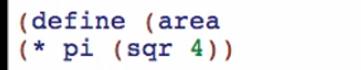


Point of variance – this is where only the function vary

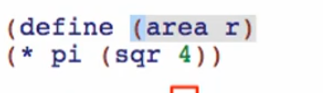
1. Take one of the expressions



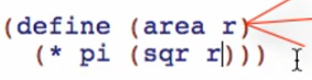
1. Wrap a function definition around with a more general name



1. Pick a parameter name for the point of variance/varying position

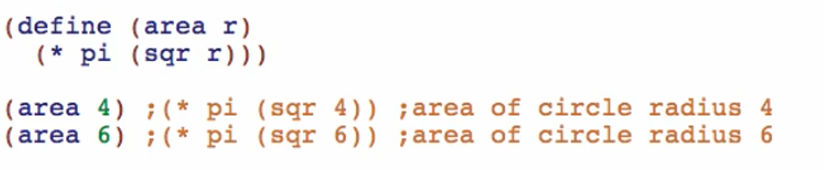


1. Replace the point of variance/varying position using the parameter

****

more “abstract” = more “general”

1. Use the function definition



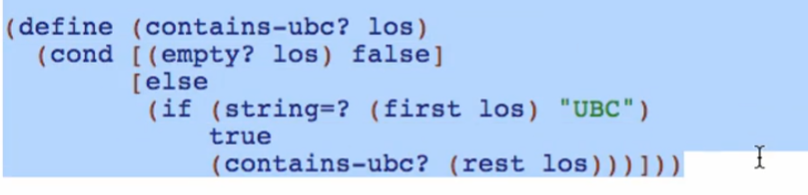
Pass the varying values as arguments for the parameter r

Again with the steps:

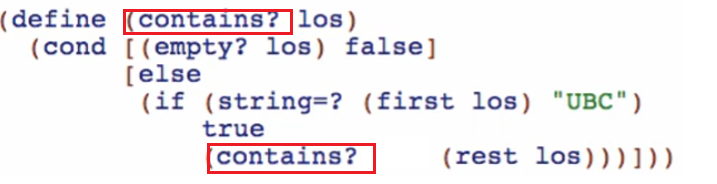
1. Identify two highly repetitive expressions



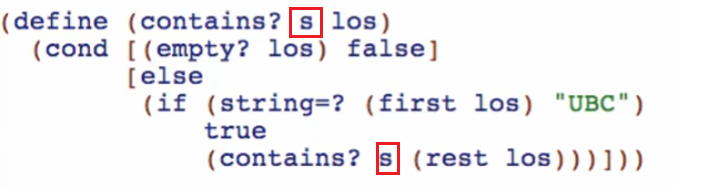
1. Take one of the functions



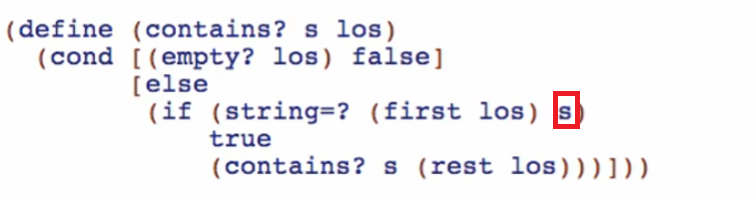
1. Since this is now a function, not an expression, no need to wrap again in a function definition. Just think with a more general name



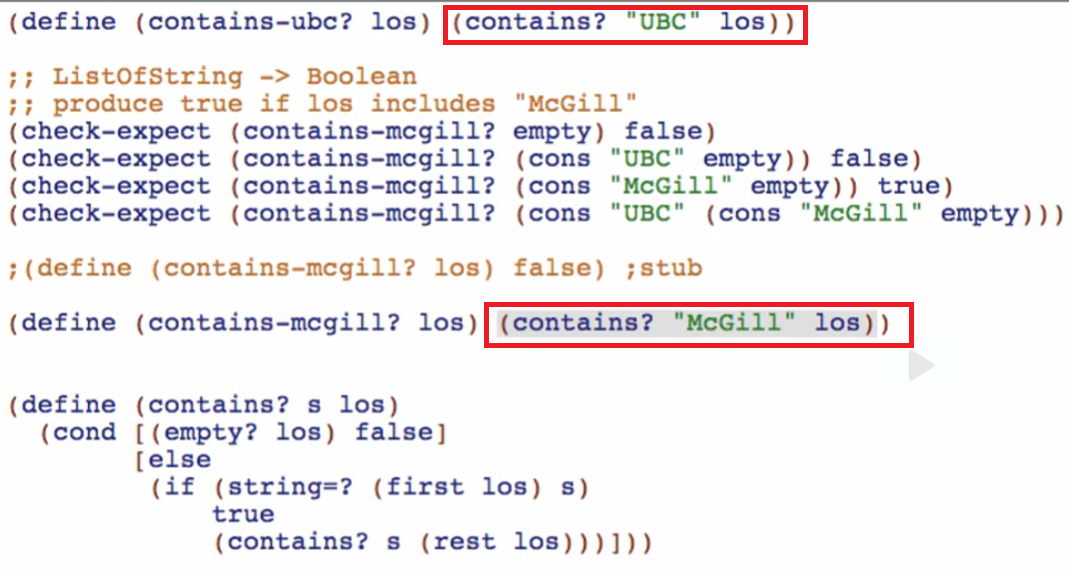
1. Pick a parameter name for the point of variance/varying position



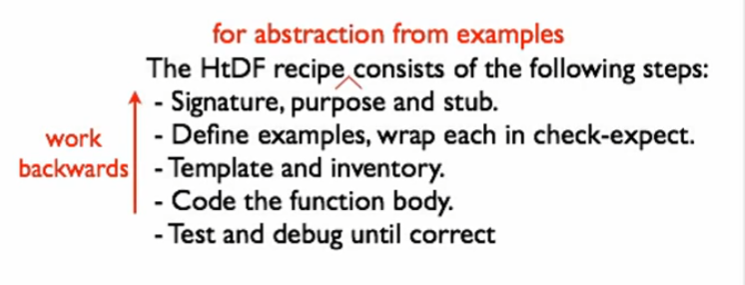
1. Replace the point of variance/varying position using the parameter



1. Use the function definition



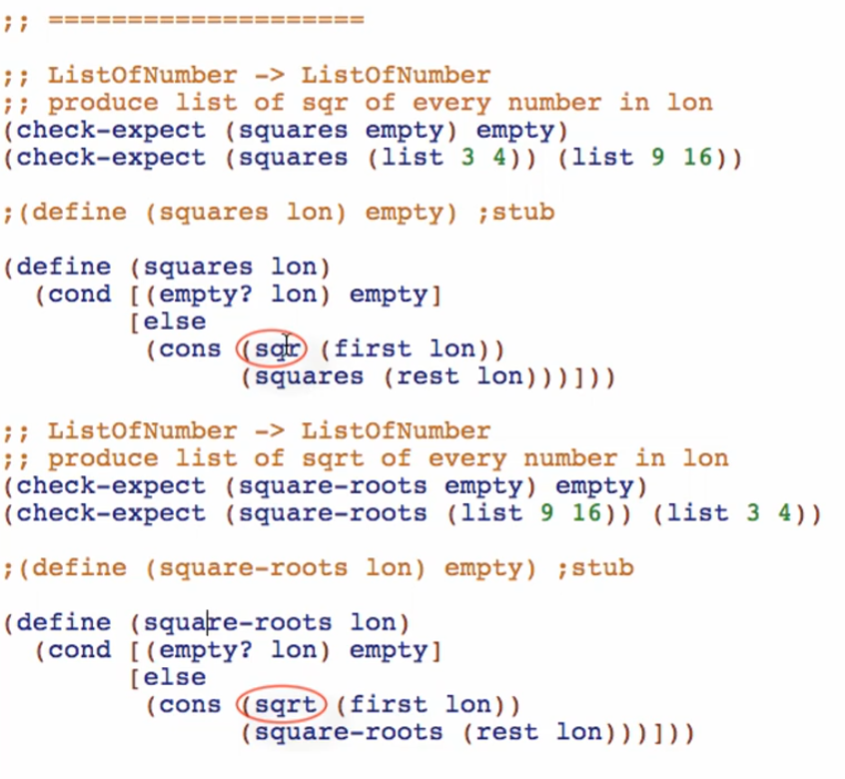
Abstraction from Examples



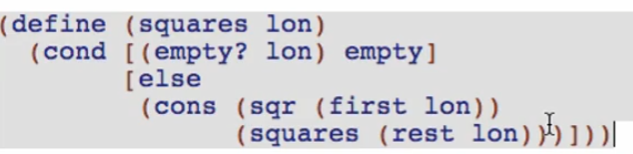
* We use abstraction to a working/existing code

Another example

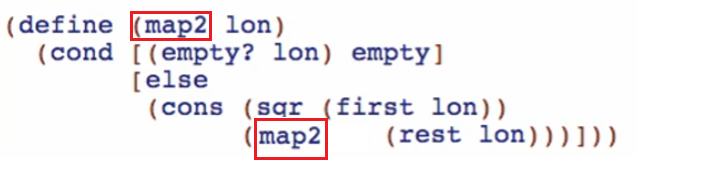
1. Identify two highly repetitive expressions and their point of variance



1. Take one of the functions

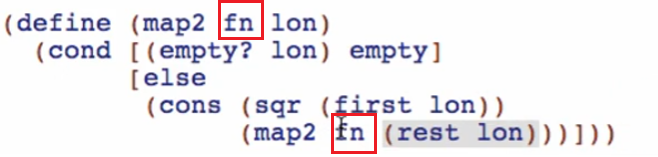


1. Since this is now a function, not an expression, no need to wrap again in a function definition. Just think with a more general name

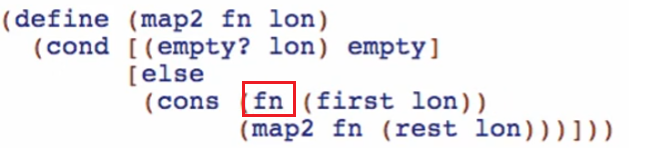


1. Pick a parameter name for the point of variance/varying position

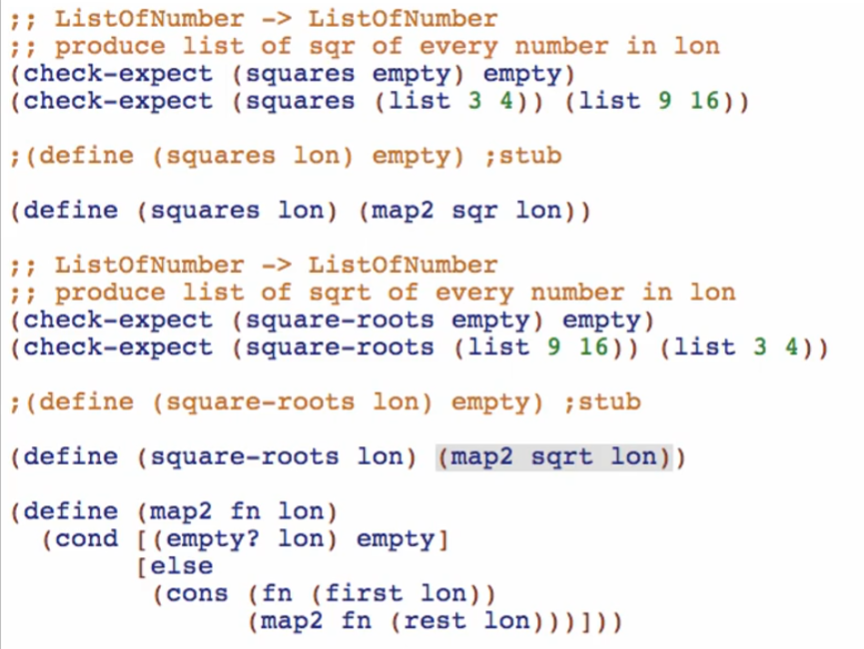
And pass the parameters added if necessary



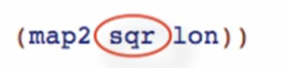
1. Replace the point of variance/varying position using the parameter



1. Use the function definition



When a function has parameter that takes another function



This is called a higher-order function

* Consume one or more functions
* Produce a function