

Quiz #6 Answers

1. Describe two ways that you can hide implementation details in ML.

We discussed four ways that you can do this in class. One is to use a signature with a structure and leave some functions or data declarations out of the signature. Similar to this is to use an opaque signature which, by default, hides the datatypes in a structure. A third way is to use abstype to create an abstract type. This is like a datatype, but only a specified set of functions will be able to see the actual type declaration. The fourth way is with local, where any number of declarations can be made in the first part of local and they are only visible to the functions in the second half while the functions in the second half are visible anywhere.

2. What is a continuation? What do they allow you to do?

A continuation is a control structure that allows a program to return to a location from earlier in its execution. The way it works is to save off the context of a particular point in the calculation and pass down information that allows the program to return to that context. With continuations you can have programs “return up” multiple levels at once. You can also implement loops and coroutines using continuations.

Extra Credit: On the back, in your favorite functional language, write a function to solve the 0/1 knapsack problem. This is a function, `maxValue` that takes a list of 2-tuples and a max “weight”. The first element of the tuple is the “value” of the item and the second is the “weight” of the item. You return the maximum value of items you can get that doesn't exceed `maxWeight`.

```
fun maxValue nil maxWeight = 0
| maxValue ((v,w)::t) maxWeight =
  if w<=maxWeight then
    Int.max(v+maxValue t (maxWeight-w),maxValue t maxWeight)
  else maxValue t maxWeight
```