

Bonus lab 11: smart pointers

- Not too demanding in programming terms, mostly a chance to actually look at smart pointers in action
- You're taking a program that currently uses smart pointers, updating them to be templated, and making sure they get instantiated ok
- The program allocates a pool of dynamically allocated resources, currently key-value pairs (defined in the `KVPair` class), each resource item tracked by smart pointer
- Functions can request/release resources from the pool, and are passed (smart) pointers to the items if available

Repo content

- README with instructions (as usual)
- pool.h, pool.cpp, lab11.cpp containing original version of program, using KVPair as the resource type
- makefile

Objective

- you're updating the program so that the resources can be anything, i.e. using templated class for Pool
- you're testing your templated version using KVPair to check that your templates are working and it's instantiating correctly
- Note that README refers to both shared and weak pointers, IIRC the code only actually uses shared ptrs