Class exercise: Refactoring (for pairs)

COMP 40

November 9, 2011

Pair

One student:

Another student:

Simplifying code

Here is a version of `Bitpack_newu`, taken from an actual submission, that weighs in at 21 lines:

```
uint64_t Bitpack_newu(uint64_t word, unsigned width,
                      unsigned lsb, uint64_t value) {
  if (width + lsb > 64) {
    RAISE(Bitpack_Overflow);
  }

  if(!Bitpack_fitsu(value, width)) {
     RAISE(Bitpack_Overflow);
  }

  uint64_t one = 1;

  uint64_t filter = shift_left64(one, width) - 1;
  filter = shift_left64(filter, lsb);

  uint64_t result = word & (˜filter);
  uint64_t insert = shift_left64(value, lsb);
  result = result | insert;

  return result;
}
```

Write a version of *this* computation that is *simpler*, *shorter*, and *easier to read*. I especially commend to you improvements that make names more meaningful and that reduce the *number* of names that the programmer has to understand. (Also note that you can refactor the code without knowing what `shift_left64` does.)