122COM: Hashing

David Croft
david.croft@coventry.ac.uk

April 11, 2016

Abstract

This week we are looking at implementing automated unit testing for your programs.

1 Introduction

Testing is a vital part of writing software. It is, however, time consuming and often dull. This week you will be looking at writing automated unit tests to check that the code you wrote in previous weeks is working as expected.

This week will be taught in Python and C++. It should be clear by this point which language you are expected to use for your answers.

The source code files for this lab are available at:

```
https://github.com/covcom/122COM_hashing.git
```

```
git@github.com:covcom/122COM_hashing.git
```

1.1 CRC32

There is very little provided code for this lab, for the most part you are expected to start from scratch although you may find it helpful to refer back to your work from previous weeks.

You are not expected to write you own hash functions, just used one of the many existing ones. CRC32 is not the best choice for a hash table but implementations of it are very easy to find and for our purposes it's fine.

Examples of using one of the many existing implementations in both Python and C++ are located in the lab_hash files.

2 Hash tables

Pre-lab work:

1. Using everything that you have learnt over the course of this module. Write a hash table class.
2. At minimum we should be able to:
   - Add key-value pairs to the hash table.
   - Number of elements in the table.

Lab work:

3. At minimum we should be able to:
   - Return associated value for a given key.
4. Additional feature could include:
   - Remove elements.
   - Deal with collisions.

Extended work:

5. Additional feature could include:
   - Automatically resize the hash table as more elements are added to reduce collisions.
6. Write a series of automated unit tests to check that your table works correctly.
   - If you want to collaborate on this step then feel free, i.e. team up and write unit tests for an abstract hash table and then see if everyone's implementations meets it.
   - You will want to agree on the names and parameters of your hash table class's member functions before hand.