# Working with open data using files & records

## Visit <http://www.tinyurl.com/GlowCompSci> click on the Higher Folder in Documents and then Open Data Files and Records. You will need your Glow Login and Password.

## Overview

*Open data* is an increasingly common phenomenon where large datasets are made available to the public, usually in the form of XML or flat (e.g. CSV) files. This activity makes use of the open data provided by the Food Standards Agency Scotland under the Food Hygiene Information Scheme. It consists of a series of exercises to explore the raw data, to identify questions that could be answered using the data, and to examine existing and develop new short "data-crunching" programs to answer the questions. The skills and understanding learned in this activity should be transferrable to any open data repository.

## Suitable for

Higher Computing Science students who already have a basic understanding of records and file handling.

## Key concepts

Open data

File formats, e.g. CSV

File handling

Records

## Learning outcomes

* Understand how a program can process complex structured data held in text files in order to derive specified information
* Appreciate the similarities and differences of arrays of records within a programming language and a database table
* Have an awareness of the phenomenon of open data and the value it has

## Success criteria

* I can understand and write programs to read in structured data from typical text file formats, e.g. CSV.
* I can design data structures using arrays and records to store structured data, e.g. an array of records
* I know how to traverse a complex data structure in order to retrieve relevant information

## Time required

3 single periods or one or two double periods

## Preparation

1. The code examples are given in Haggis. You can use them as is with your pupils, or else translate them into the language of your choice. Over time, we will populate the Glow site with translations to the common languages.
2. This can be a paper/desk exercise, or else pupils can work with the data file and programs at a machine. If you opt for the latter, you will need to download the CSV open data file as well as the example programs in your own language and make them accessible to your pupils.
3. If you are working on paper, then Handout 1 is copied onto A3 and then cut in two on the long axis.
4. Working in groups of 2 or 3, ensure each group has one copy of the handouts.

## Prior learning assumed

National 5 SDD outcomes, and a basic working knowledge of records and files. Able to design solutions to simple problems.

# Outline of Activity

* Students will work in groups of 2-3 for this task. The activities are graded in difficulty with a blend of code comprehension with worked out solutions to problems, problem solving and code writing.
* Students first imagine of what use open data might be.
* They then explore the example data file from the Food Standards Agency Scotland, deciding on some analyses of the data.
* For a particular set of analyses, they determine what data is required from the file
* They review a program for reading in this data, and then write a plan to carry out a particular analysis
* They compare their plan against a provided solution
* They then solve a small problem from scratch, and then a larger, more challenging problem. Solutions are available for both these problems.

## Introduction

In groups of 3 or 4 discuss: what data you think governments, or government agencies, have collected that you'd like to explore? Possible examples

* School inspection reports
* Exam statistics by school
* Driving test results by centre
* Food standards information
* Health related stats?
* Voting stats of young people

Governments are increasingly being required to make this kind of data available on-line – it's called open data. It's early days, but the quantity will inevitably grow.

"So what?", you may ask. "What can I do with a huge pile of data?" With a little knowledge and some additional programming skills, you will be quickly able to manipulate this data to get answers to questions that interest you – in just a few minutes. And the same skills will enable you to analyse data coming from any source, such as you might find as a finance worker, scientist, researcher, and so on.

We're going to explore this area using a particular set of open data drawn from Food Standards Scotland, reporting on food outlet inspections for Glasgow, stored in a CSV file. Equivalent data is available for 30 of the 32 local authorities in Scotland.

## Activity

1. Hand out the half A3 sheet (half of Handout 1) with a sample of the data file on it – header row and 10-20 rows of data. In groups, eyeball the data, and work out what's in there. What is the structure? Can you identify entities and their attributes?

* *There could be Business, Rating, Local Authority – it depends rather on how you choose to interpret the raw data.*

1. What questions can you answer with this data?

* *How many failed in my postcode, within a radius of my current position, in the last n days? What are their names? List all the types of food outlet. Count of restaurants near here. Which post-code area (e.g. G12, G4) has the highest percentage of failed outlets at this time?*

1. Say we want to answer the following questions? What's the minimum data from the larger data set in the file that we need to do this?

* How many failed in my postcode, within a radius of my current position, in the last n days?
  + What are their names?
* List all the types of food outlet.
* Count of restaurants near here.
* Which post-code area (e.g. G12, G4) has the highest percentage of failed outlets at this time?
* *Business name, business type, postcode, rating date, rating result, location*

1. Let's explore how to read that data into a suitable data structure in our program. Review the program in Handout 2. In particular, make sure you can find and understand the following:

* The record type declaration
* Where the file is opened and how lines are read in
* How the data is extracted from each line and placed in a record
* How the whole data set is stored

1. Given that you've read the data in as above, let's work out how to process the data to get the name of all failed outlets within a 1 mile radius of a given position (e.g. my current position). In your group, develop a plan for how you would go about answering the question.
2. Now review the code in Handout 3 for solving this problem. How well does it match your solution? To help with understanding it, annotate each line with two items:

* The construct being used and how it works
* How the line contributes to solving the problem

1. Now write code to count up how many outlets passed in the G12 postcode area.

* *A solution is on Handout 4 (though you may not want to hand it out!)*

1. Now write code to answer the following question: Which post-code area (e.g. G12, G4) has the highest percentage of failed outlets at this time? Consider the following:

* Assume you have the code that reads the file into records
* You will need to traverse the entire array of records
* For each unique postcode area you find, you need a new record type value, holding the post-code area, and summary data about the outlet ratings in that area. (what summary data do you need to store in order to work out the percentage of failed outlets?)
* As you traverse the existing array of records, you need to update the information you are storing for each post-code area.
* *The following is a rough plan for the code*
  + *Define a record (post-code area, number of failed outlets, total number of outlets)*
  + *Set up a data array of this record type*
  + *Access each of the records in the main data structure in turn:*
    - *the data array must be checked to see if the record's post-code area has been seen before*
    - *If it's a new post-code area*, *a new entry must be created in the data array, otherwise the existing entry can be updated.*
  + *Finally, the data array must be traversed, calculating the percentage of failed outlets in each post-code area, and keeping a link to the entry in the data array with the largest percentage.*
* *A complete solution is in Handout 4.*

1. A closing question. If you wanted to answer just one simple question of the data file – say, get the number of failed outlets in the entire local authority – do you need to go to the trouble of constructing the full array of records, as we did here, or can you write a simpler program? If so, give an outline plan of such a program.

* *A solution is given in Handout 4.*

## Review

Pupils should reflect on the similarities and differences between databases and programming languages with respect to manipulating this kind of data.

How easy would it have been to take the CSV file provided and use a database system to analyse the data?

Having worked with records, can you see other application contexts where they might be valuable? Where might you use them in a game, for example?

# Suggested follow up work

* Students can investigate what other open data sets are available. One example would be to find the open data on food standards for their own local authority.
* A next step is to consider not just reading in and processing data from these data sets, but keeping their own file of gleaned data. This requires writing the complex data structures they created back out into files – and then reading them in again as required.

# Acknowledgements AND License

This idea and associated resources were developed by



Professor Quintin Cutts – University of Glasgow

Professor Richard Connor – University of Strathclyde

Kelly MacDonald – Harris Academy- Dundee City

Kirsty Allan – Greenwood Academy- North Ayrshire

During a Craft the Curriculum for Higher Computing Science event

A joint event held by PLAN C and Education Scotland







© Crown copyright 2015. You may re-use this information (excluding logos) free of charge in any format or medium, under the terms of the [Open Government Licence](http://www.nationalarchives.gov.uk/doc/open-government-licence/).

Where we have identified any third party copyright information you will need to obtain permission from the copyright holders concerned.