

Cold Call Helper

Programmer's Documentation

Authors:

Nolan Cassidy, Andrew Evelhoch, Jake Gianola, Kyle Kincaid, Kylie Quan

Date last updated: 4/95/2019

This document contains an in-depth description of the programming of the features of the Cold Call Helper. Anyone wishing to add, remove or change functionality should read this first.

Table of Contents:

1. General composition
2. Description of modules
 - a. GUI/Interface - (ColdCallHelper.py)
 - b. Queue - (class_Queue.py)
 - c. Data Repository - (repository.py)
 - d. Export Daily/Term Summaries - (termsum.py & dailysum.py)
3. Files used by program

The ColdCall directory holds five python files and a roster.tsv. When the ColdCallHelper is executed it compiles the student information from the roster and displays it on screen for the teacher to be able to call on students.

==== 1. General Composition =====

Cold Call Helper is composed of three different classes, in a strict hierarchy. The top level class, Interface creates and holds an instance of the class_Queue class, and the class_Queue class creates and holds an instance of the data repository, databoard. Starting at the bottom to the top, here is the concerns of each class:

The data repository is the bottom class. Its concerns are reading data in, storing it, altering it and saving it. If you wish to make changes to how the data is stored, change this class.

The second class up is the queue. Its concerns are making sure the students are called on in a random order, and fairly. It also gets data from the data repository to expose to the GUI, and allows the GUI to tell the data repository what operations to do. If you wish to make changes to how the students are randomized, if you need to see more or less data from the repository in the GUI, or if you need to add more ways for the GUI to alter the data repo, change this class.

The top class is the <GUI>. Its concerns are displaying a GUI and handling user input. If you wish to make changes to what is displayed on screen,

