```
/****
 *
 * Module main.C contains the main C++ function for the Rolodex system.  It
 * constructs instances of the model, view, and process classes for the Rolodex
 * and gives control to the view, which will manage communication with the end
 * user.
 *
 * This is an initial implementation for GUI testing purposes, with no process
 * classes defined.
 *
 */

#include "iv-support.h"
#include "quit-exception.h"
#include "rolodex-menu-ui.h"
#include "rolodex-tool.h"
#include "rolodex.h"


void CleanUp(Screen* s, RolodexTool* rt, RolodexUI* rui);
/*
 * Destruct top-level objects;
 */

main() {
    Screen* s;                  // The GUI screen
    RolodexTool* rt;            // The top-level Rolodex model
    RolodexUI* rui;             // The top-level Rolodex gui

    /*
     * Construct the GUI screen, thereby initializing the GUI system.  In this
     * InterViews-based implementation, the GUI screen contains an InterViews
     * World, which is the top-most object in an InterViews GUI.
     */
    s = new Screen();

    /*
     * Construct the top-level Rolodex model.  It will in turn construct all
     * subsidiary model classes and the data objects they require.
     */
    rt = new RolodexTool(NULL);

    /*
     * Construct and compose the Rolodex GUI.  Compose will lay out all GUI
     * components, including dialog boxes that appear during the course of user
     * interaction.
     */
    rui = new RolodexMenuUI(s, rt);
    rui->Compose();

    /*
     * Store the view in the model to enable two-way communication between the
     * model and the view.
     */
    rt->SetView(rui);

    /*
     * Insert the view’s window into the UI screen.
     */
    rui->Show();

    /*
     * Run the view, which will assume full control by entering the InterViews
     * GUI event loop.  Control will only return here when the user issues a
     * quit command.  The response to the quit command is to throw the
     * QuitException.
     */
    try {
        rui->Run();
    }
    catch (QuitException*) {
        /*
         * Clean up by deleting all of the constructed objects.
         */
        CleanUp(s, rt, rui);
    }
}

void CleanUp(Screen* s, RolodexTool* rt, RolodexUI* rui) {
    delete rt;
    delete rui;
    delete s;
}
```