```
/*
 *  ControlModule.h
 *  Defines the Control Module class, which handles the flow of data between the consumer
 *      and the action modules
 *
 *  Created: 1/3/2017 by Ryan Tedeschi
 */

#ifndef CONTROLMODULE_H
#define CONTROLMODULE_H

/*
 *  Control Module Input parameter type placeholders
 */
#define SOURCE_LANGUAGE string
#define MODULE_ID string
#define CODE_INPUT vector<string>
#define FUNCTION_ARGS vector<arg>
#define MODULE_RESPONSE CASP_Return*
#define LANGUAGE_DESCRIPTOR_OBJECT LanguageDescriptorObject*
#define MARKUP_OBJECT Markup*
#define CODE_OUTPUT vector<string>
#define MODULE_REF CASP_Plugin*

#include "../shared/CASP_Plugin/CASP_Plugin.h"
#include "../plugins/plugins.h"
#include <string>
#include "../shared/LanguageDescriptor/LanguageDescriptor.h"
#include "../shared/Markup/Markup.h"

using namespace std;

class ControlModule {
    public:
        ControlModule();
        ~ControlModule();
        void Run(SOURCE_LANGUAGE, MODULE_ID, CODE_INPUT, FUNCTION_ARGS);

    private:
        LANGUAGE_DESCRIPTOR_OBJECT GetLanguageDescriptor(SOURCE_LANGUAGE) throw (std::string);
        bool ValidateSourceLanguage(SOURCE_LANGUAGE);
        LANGUAGE_DESCRIPTOR_OBJECT ReadLanguageFile(SOURCE_LANGUAGE) throw (std::string);
        CODE_OUTPUT CoalesceCode(CODE_INPUT); // is this necessary?..
        MARKUP_OBJECT Parse(CODE_OUTPUT, LANGUAGE_DESCRIPTOR_OBJECT);
        void Execute(MARKUP_OBJECT, LANGUAGE_DESCRIPTOR_OBJECT, MODULE_ID, FUNCTION_ARGS);
        MODULE_REF ModuleRetrieval(MODULE_ID);
        void ModuleExecution(MODULE_REF, MARKUP_OBJECT, LANGUAGE_DESCRIPTOR_OBJECT, FUNCTION_ARGS);
        void FormatOutput();

        CASP_Return* returnData = new CASP_Return();
};

#endif
```