

Section 8 (Week 9) – STARTER CODE

Problem and solution authors include Jerry Cain and Aubrey Gress.

Problem 2 Starter Code: Using Polymorphism with JavaScript Object Notation

```
/**
 * json-lite-inheritance.cpp
 * -----
 * Provides the test framework to illustrate how inheritance works.
 */

#include "console.h"
#include "vector.h"
#include "map.h"
#include "strlib.h"
#include "tokenscanner.h"
#include "error.h"
#include "simpio.h"
using namespace std;

class JSONElement {
public:
    virtual ~JSONElement() {};
    virtual string toString() = 0;
private:
};

class JSONInt : public JSONElement {
public:
    JSONInt(int i) { value = i; }
    ~JSONInt () {}
    string toString() { return integerToString(value); }
private:
    int value;
};

JSONElement *parseJSON(TokenScanner& scanner) {
    string lookahead = scanner.nextToken();
    if (lookahead.empty()) return NULL;

    if (isdigit(lookahead[0])) {
        return new JSONInt(stringToInteger(lookahead));
    } else {
        error("JSON element type passed to parseJSON not yet supported.");
    }

    // should never get to the bottom here, but compiler can't figure that out
    return NULL;
}
```

```
int main() {
    setConsoleSize(1000, 700);
    setConsoleFont("Courier-BOLD-18");
    TokenScanner scanner;
    scanner.ignoreWhitespace();
    scanner.scanStrings();

    ifstream infile("json-lite-data.txt");
    scanner.setInput(infile);

    while (true) {
        try {
            JMLElement *jsonRoot = parseJSON(scanner);
            if (jsonRoot == NULL) break;
            cout << jsonRoot->toString() << endl;
            delete jsonRoot;
        } catch (ErrorException& ex) {
            cout << ex.getMessage() << endl;
        }
    }

    cout << endl;
    getline("Everything's been read in and printed out. Press ENTER to quit.");
    return 0;
};
```