```cpp
#include "CASP_Return.h"

void GenericData::Print() {}

GenericObject::GenericObject() {}
GenericObject::GenericObject(unordered_map<string, GenericData*> map) {
    data = map;
}
void GenericObject::Print() {
    int count = 0;
    cout << "{";
    for (auto it = data.begin(); it != data.end(); ++it ) {
        if (count++ > 0) {
            cout << ",";
        }
        cout << "\"" << it->first << "\":";
        if (it->second != NULL)
            it->second->Print();
        else
            cout << "null";
    }
    cout << "}";
}

void GenericObject::Add(string key, GenericData* d) {
    data[key] = d;
}

GenericData* GenericObject::At(string key) {
    return data[key];
}


GenericArray::GenericArray() {}
GenericArray::GenericArray(vector<GenericData*> list) {
    data = list;
}

void GenericArray::Print() {
    cout << "[";
    for (int i = 0; i < data.size(); i++) {
        if (i > 0)
            cout << ",";
        data[i]->Print();
    }
    cout << "]";
}

void GenericArray::Add(GenericData* d) {
    data.push_back(d);
}

GenericData* GenericArray::At(int index) {
    return data[index];
}

CASP_Return::CASP_Return() {
    Add("Data", new GenericObject());
    Add("Warnings", new GenericArray());
    Add("Errors", new GenericArray());
}

GenericArray* CASP_Return::Errors() {
    return (GenericArray*)data["Errors"];
}
GenericArray* CASP_Return::Warnings() {
    return (GenericArray*)data["Warnings"];
}
GenericObject* CASP_Return::Data() {
    return (GenericObject*)data["Data"];
}
void CASP_Return::AddStandardWarning(string message, int warningId) {
    GenericObject* warn = CreateObject({ { "id", CreateLeaf(warningId) }, { "message", CreateLeaf(message) } });
    Warnings()->Add(warn);
}
void CASP_Return::AddStandardError(string message, int errorId) {
    GenericObject* err = CreateObject({ { "id", CreateLeaf(errorId) }, { "message", CreateLeaf(message) } });
    Errors()->Add(err);
}


GenericObject* CreateObject() {
    GenericObject* ob = new GenericObject();
    return ob;
};
GenericObject* CreateObject(unordered_map<string, GenericData*> map) {
    GenericObject* ob = new GenericObject(map);
    return ob;
};
GenericArray* CreateArray() {
    GenericArray* arr = new GenericArray();
    return arr;
};
GenericArray* CreateArray(vector<GenericData*> list) {
    GenericArray* arr = new GenericArray(list);
    return arr;
};
```