

```
//
// File: variable.h
//
//
//Massachusetts Institute of Technology
//16.412J/6.834J Cognitive Robotics
//
//Russian Doll Search
//
//Problem Set #2
//Due: in class Wed, 3/9/05
//
//Lawrence Bush, Brian Bairstow
//{ bush12, bairstow }@mit.edu
//
// -----
//
// variable.h - Contains a variable class.
//             variable object stores variable information,
//             and performs formatted output and other
//             functions on variable data.

#ifdef _variable_h_
#define _variable_h_
#include <string>
#include <iostream>
#include <fstream>
using namespace std;
//
class variable {

    friend std::ostream& operator << ( std::ostream& ostr,
        const variable& v );

public:
    variable(){}; // default constructor --

    // Constructor - assigns all variable attributes
    variable( int var_index_in, int domain_size_in, int domain_value_in )
        : var_index(var_index_in), domain_size(domain_size_in), domain_value
        (domain_value_in)
    {

    }

    //*****
    //*****
    // Get functions to retrieve a statistic about the variable.
    //*****
    //*****

    int get_domain_value() const {return domain_value;} // returns variable value
    int get_var_index() const {return var_index;} // returns variable value
    int get_domain_size() const {return domain_size;} // returns variable value

    bool increment_domain_value() {
        domain_value++;
        if(domain_value >= domain_size) {
            domain_value--;
            return false; // check for going to far
        } else {
            return true;
        }
    }

    bool max_domain_value() const {return domain_value==domain_size;} // returns variable
    value
```

```
bool next_domain_value(){
    domain_value++;
    return max_domain_value();
} // returns variable value

private:
//*****
//*****
//      Private Data Members
//*****
//*****
int var_index;
int domain_size;
int domain_value;

};

//*****
// Overloaded non-member << operator
//*****
std::ostream &
operator << ( std::ostream & ostr, const variable & v )
{
    ostr <<"Variable Index: " << v.get_var_index() << ", Domain Size: " << v.
    get_domain_size() << ", Domain Value: " << v.get_domain_value() << endl;
    return ostr;
}

#endif
```