```
/****
 *
 * Class Card is the model class for the Card object.  A Card contains the
 * basic information for a person stored in a Rolodex: name, SSN id, age, sex,
 * and street address.
 *
 */

#ifndef cardIncluded
#define cardIncluded

#include "basic-types.h"
#include "model.h"


class Card : public Model {
  public:
    Card(Name* n, Id id, Age age, Sex s, Address* addr);
    /*
     * Construct a card with the given field values.  Note that no validity
     * checking of the given values is performed here.  Rather, input value
     * checking is done in the Rolodex::Add function, q.v.
     *
     * post: (this->n == n) and (this->id == id) and (this->age == age) and
     *       (this->s == s) and (this->addr == addr);
     */

    ~Card();
    /*
     * Deep delete this by deleting all non-atomic field values.
     *
     * post: freed(this) and freed(n) and freed(addr);
     */

    /**
     * Field access  functions.
     */
    Name* GetName();
    /*
     * Return the name of this.
     *
     * post: return == n;
     */

    Id GetId();
    /*
     * Return the id of this.
     *
     * post: return == id;
     */

    Age GetAge();
    /*
     * Return the age of this.
     *
     * post: return == age;
     */

    Sex GetSex();
    /*
     * Return the sex of this.
     *
     * post: return == s;
     */

    Address* GetAddr();
    /*
     * Return the name of this.
     *
     * post: return == addr;
     */

    bool Equal(Card* c);
    /*
     * Return true if this is deep equal to the given card , i.e. card all of
     * the fields of this are equal to all of the fields of the given card.
     *
     * post: (this->n == c->n) and (this->id == c->id) and
     *       (this->age == c->age) and (this->s == c->s) and
     *       (this->addr == c->addr);
     */

    void Print();
    /*
     * Print this to stdout.  In the current design, the specs are the same as
     * Dump below, q.v.  In general, if there is a distinction between Print
     * and Dump it is that the former produces a more nicely formated output
     * than the latter.
     */

    void Dump();
    /*
     * Dump this to stdout.
     *
     * post: (stdout == "Name: " + n + "\n" +
     *                  "Id: " + id + "\n" +
     *                  "Age: " + age + "\n" +
     *                  "Sex: " + s + "\n" +
     *                  "Addr: " + addr + "\n\n")
     */


  protected:
    Name* n;                    // Person name and record search key
    Id id;                      // Person id and unique record key
    Age age;                    // Person age
    Sex s;                      // Person sex
    Address* addr;              // Person street address

    Card(Card &);               // Standard protected copy constructor
    Card& operator=(Card&);     // Standard protected op =
};
```

```

#endif
```