Coms 262 **The `State` Class** **Spring 2018**

Data Members

The `State` class has seven data members.

- `string m_abbr;` – The states's abbreviation.
- `string m_name;` – The states's name (with embedded blanks).
- `string m_ capital;` – The states's capital city (with embedded blanks).
- `int m_pop;` – The states's population.
- `int m_rep;` – The states's representation in the House of Representatives.
- `Date m_joined;` – The date when the state joined the union.
- `double m_area;` – The states's land area (in mi^2).

Member Functions

Constructors

- `State();`

  Constructs a `State` object in which all data members take on their default values.

- `State(const string& ab, const string& nm, const string& cc, int pp, int rp, const Date& jn, double ar);`

  Constructs a `State` object with abbreviation `ab`, name `nm`, capital city `cc`, population `pp`, number of representatives `rp`, date when it joined `jn`, and land area `ar`.

- `State(string ab);`

  Constructs a `State` object with abbreviation `ab`. All other data members will take on their default values.

Inspectors

- `string abbr() const;`

  Returns the abbreviation.

- `string name() const;`

  Returns the name.

- `string capital() const;`

  Returns the capital city.

- `int pop() const;`

  Returns the population.

- `int rep() const;`

  Returns the number of representatives.

- `Date joined() const;`

  Returns the date when the state joined the union.

- `double area() const;`

  Returns the land area.

Mutators

- `void abbr(const string& ab);`

  Sets the abbreviation to `ab`. Verify that `ab` has length 2.

- `void name(const string& nm);`

  Sets the name to `nm`.

- `void capital(const string& cc);`

  Sets the capital city to `cc`.

- `void pop(int pp);`

  Sets the population to `pp`. Verify that `pp` is not negative.

- `void rep(int rp);`

  Sets the number of representatives to `rp`. Verify that `rp` is at least 1.

- `void joined(const Date& jn);`

  Sets the date when the state joined the union to `jn`.

- `void area(double ar);`

  Sets the area to `ar`. Verify that `ar` is not negative.

Facilitators

- `void input(istream& in);`

  Reads an `State` object. The format of an `State` object is

  abbr name\t*capital*\t*population representation joined area*

  where \t is a tab character. The other fields are separated by blanks.

- `void output(ostream& out) const;`

  Writes an `State` to the output stream. The output format is the same as the input format.

- `bool isEqual(const State& st) const;`

  Determines whether two `State` objects are equal. Two `State` objects are equal if and only if they have the same abbreviation.

- `bool isLessThan(const State& st) const;`

  Determines whether one `State` object is less than another `State` object. One `State` object is less than another `State` object if and only if the *name* of that state is less than the *name* of the other state, in alphabetical order.

Other Member Functions

- `void display(ostream& out) const;`

  Displays the data of a State in the form

  `State:` *name*
  `Abbreviation:` *abbreviation*
  `Capital city:` *capital city*
  `Population:` *population*
  `Representation:` *number of representatives*
  `Admitted to union:` *date joined*
  `Area:` *area*

Non-member Operators

- `istream& operator>>(istream& in, State& st)`

  Reads a `State` object from the input stream.

- `ostream& operator<<(ostream& out, const State& st)`

  Writes a `State` object to the output stream.

- `bool operator==(const State& st1, const State& st2)`

  Determines whether two `State` objects are equal.

- `bool operator!=(const State& st1, const State& st2)`

  Determines whether two `State` objects are not equal.

- `bool operator<(const State& st1, const State& st2)`

  Determines whether one `State` object is less than another `State` object.