Mask Class Documentation

Source File: Mask.h
Class Header: `class Mask : public Object`

Overview

The *Mask* class is a comparable container class for an inaccessible digit between 1 and 9 inclusively that can be voided.

Constructors

- `Mask()` (default constructor)
  - **Purpose:** Randomly assigns the mask a digit between 1 and 9 inclusively.
- `Mask(const Mask& obj)` (copy constructor)
  - **Purpose:** Constructs a deep copy of *obj*.
  - **Parameter(s):**
    - *obj*: Constant *Mask* reference object.

Destructor

- `~Mask()`
  - **Purpose:** Does nothing.

Assignment Operators

- `operator=(const Mask& rhs)`
  - **Purpose:** Constructs a deep copy of *rhs*.
  - **Parameter(s):**
    - *rhs*: Constant *Mask* reference object.
  - **Return:** `*this`.

Methods

- `Void()`
  - **Purpose:** Voids the mask value.
- `IsVoid() const`
  - **Purpose:** Checks if the mask value is voided.
  - **Return:** It returns true if the mask value is voided; otherwise, it returns false.
- `ToString() const override`
  - **Purpose:** Provides a string representation of the *Mask* object.
  - **Return:** A string `"O"` if the mask value is not voided; otherwise, a string `"X"`.

Non-Member Functions

- `operator==(const Mask& lhs,const Mask& rhs)`
  - **Purpose:** Checks if the mask values of *lhs* and *rhs* are equal.
  - **Parameters:**
    - *lhs*: Constant reference of an *Mask* object.
    - *rhs*: Constant reference of an *Mask* object.
  - **Return:** It returns true if the mask values of the parameters are equal; otherwise, it returns false.

- `operator!=(const Mask& lhs, const Mask& rhs)`
  - **Purpose:** Checks if the mask values of *lhs* and *rhs* are different.
  - **Parameters:**
    - *lhs*: Constant reference of an *Mask* object.
    - *rhs*: Constant reference of an *Mask* object.
  - **Return:** It returns true if the mask values of the parameters are different; otherwise, it returns false.

Example:

```
#include <iostream>
#include "Mask.h"

int main()
{
  srand(time(nullptr));
  Mask a, b;

  if(a == b)
  {
    std::cout << "They match\n";
  }
  else
  {
    std::cout << "They do not match\n";
  }
  b.Void();
  std::cout << a << b << "\n"; // Will display OX
  return 0;
}
```