

cs225f20-b.sg

[Dashboard](#) / [My courses](#) / [cs225f20-b.sg](#) / [26 October - 1 November](#) / [Assignment 4 upload](#)

- [Description](#)
- [Submission](#)
- [Edit](#)
- Submission view

Grade

Reviewed on Monday, 9 November 2020, 2:05 PM by Automatic grade

grade: A

Assessment report[-]

[-]Summary of tests

+-----+
| 25 tests run/25 tests passed |
+-----+

Submitted on Monday, 9 November 2020, 2:05 PM ([Download](#))

bitset.h

```
1  ▾ /* Start Header
2  *****/
3  ▾ /*!
4  \file  bitset.h
5  \author Goh Wei Zhe, weizhe.goh, 44000119
6  \par    email: weizhe.goh@digipen.edu
7  \date   November 9, 2020
8  \brief  Contains declaration of member functions of class bitset
9
10 Copyright (C) 2020 DigiPen Institute of Technology.
11 Reproduction or disclosure of this file or its contents without the
12 prior written consent of DigiPen Institute of Technology is prohibited.
13 */
14 ▾ /* End Header
15 *****/
16
17 #ifndef _BITSET_H_
18 #define _BITSET_H_
19
20 #include <bitset>
21
22 namespace cs225
23 {
24     template<std::size_t N>
25     class bitset
26     {
27     public:
28
29         unsigned char* _bitset;
30
31         bitset();
32
33         bitset<N>& set(std::size_t pos, bool value = true);
34         bitset<N>& reset(std::size_t pos);
35         bitset<N>& flip(std::size_t pos);
36
37         std::size_t count() const;
38         std::size_t size() const;
39         bool test(std::size_t pos) const;
40
41         std::string to_string(char c1 = '0' , char c2 = '1') const;
42         bool operator[](std::size_t pos) const;
43
44         ~bitset();
45     };
46 }
47
48 #include "bitset.hpp"
49
50 #endif
```

bitset.hpp

```

1  /* Start Header
2  *****/
3  /*!
4  \file   bitset.hpp
5  \author Goh Wei Zhe, weizhe.goh, 44000119
6  \par    email: weizhe.goh@digipen.edu
7  \date   November 9, 2020
8  \brief  Contains definition of member functions of class bitset
9
10 Copyright (C) 2020 DigiPen Institute of Technology.
11 Reproduction or disclosure of this file or its contents without the
12 prior written consent of DigiPen Institute of Technology is prohibited.
13 */
14 /* End Header
15 *****/
16
17 #ifndef BITSET_HPP_
18 #define BITSET_HPP_
19
20 #include "bitset.h"
21
22 template<size_t N>
23 cs225::bitset<N>::bitset()
24 {
25     size_t _size = (N + CHAR_BIT - 1) / CHAR_BIT;
26     _bitset = new unsigned char[_size] {0};
27 }
28
29 template<size_t N>
30 cs225::bitset<N>& cs225::bitset<N>::set(size_t pos, bool value)
31 {
32     if (value)
33         _bitset[pos / CHAR_BIT] |=
34             static_cast<unsigned char>(1U << pos % CHAR_BIT);
35     else
36         _bitset[pos / CHAR_BIT] &=
37             static_cast<unsigned char>(~(1U << pos % CHAR_BIT));
38
39     return *this;
40 }
41
42 template<size_t N>
43 cs225::bitset<N>& cs225::bitset<N>::reset(size_t pos)
44 {
45     _bitset[pos / CHAR_BIT] &=
46         static_cast<unsigned char>(~(1U << pos % CHAR_BIT));
47
48     return *this;
49 }
50
51 template<size_t N>
52 cs225::bitset<N>& cs225::bitset<N>::flip(size_t pos)
53 {
54     _bitset[pos / CHAR_BIT] ^=
55         static_cast<unsigned char>(1U << (pos % CHAR_BIT));
56
57     return *this;
58 }
59
60 template<size_t N> size_t cs225::bitset<N>::count() const
61 {
62     size_t count = 0;
63
64     for (size_t i = N; i > 0; i--)
65         (bitset<N>::test(i) ? ++count : count);
66
67     return count;
68 }
69
70 template<size_t N>
71 std::string cs225::bitset<N>::to_string(char c1, char c2) const
72 {
73     std::string s;
74
75     char zero = static_cast<unsigned char>(c1);
76     char one = static_cast<unsigned char>(c2);
77
78     for (size_t i = N; i > 0; i--)
79         s.push_back(bitset<N>::test(i) ? one : zero);
80
81     return s;
82 }
83
84 template<size_t N> size_t cs225::bitset<N>::size() const
85 {
86     return N;
87 }
88
89 template<size_t N> bool cs225::bitset<N>::test(size_t pos) const
90 {
91     if (N <= pos)
92         throw std::out_of_range("Out of range");
93
94     return (_bitset[pos / CHAR_BIT] & (1U << pos % CHAR_BIT)) != 0;
95 }
96
97 template<size_t N> bool cs225::bitset<N>::operator[](size_t pos) const
98 {
99     return (_bitset[pos / CHAR_BIT] & (1U << pos % CHAR_BIT)) != 0;
100 }
101
102 template<size_t N> cs225::bitset<N>::~~bitset()
103 {
104     delete []_bitset;
105 }
106
107 #endif // !BITSET_HPP

```

◀ Assignment 4 files

Jump to...



Attendance cs225f20-b.sg Tuesday
3/11/2020 1:30pm-3:10pm ▶

[VPL](#)

You are logged in as [Wei Zhe GOH](#) ([Log out](#))
[cs225f20-b.sg](#)
[Data retention summary.](#)
[Get the mobile app](#)